論理LSIの自動設計における標準セルとマクロセル混在LSIの配置配線手法の研究(Dissertation_全文)

AUTHOR(S):

早瀬, 道芳

CITATION:

早瀬, 道芳. 論理LSIの自動設計における標準セルとマクロセル混在LSIの配置配線手法の研究. 京都大学, 1996, 博士(工学)

ISSUE DATE:

1996-05-23

URL:

https://doi.org/10.11501/3112307

論理ＬＳＩの自動設計における
標準セルとマクロセル混在ＬＳＩの
配置配線手法の研究

早瀬　道芳

まえがき

本論文は、著者が（株）日立製作所中央研究所に在籍中、１９８０年代に従事した論理ＬＳＩの自動配置配線手法に関する研究と、１９９３年に岡山県立大学へ移った後に従事した、配線長最小問題としてのスタイナ木に関する研究と、未配線入力等に使用するエキスパートシステムの知識の記憶最小表現に関する研究をまとめたものである。

著者が論理ＬＳＩのレイアウト（配置配線）設計自動化の研究を始めたのは１９８０年である。日立製作所中央研究所では、これより先１９７０年頃より先駆的なレイアウト設計自動化の研究が始められていた。しかし、当時は論理ＬＳＩの規模が小さく、一人で論理ＬＳＩのチップ全体を見渡してレイアウトできる時代であったので、レイアウト設計自動化は実用にならなかった。しかし、１９８０年頃になって論理ＬＳＩの規模が数万ゲートになって来ると、複数の設計者による分担レイアウト設計が行われていたが、それでも設計期間の遅延が目立つようになって来た。このため、再びレイアウト設計自動化の研究が取り上げられた時、著者は論理ＬＳＩのレイアウト設計自動化の研究を始めた。

研究を始めた頃は、レイアウト設計自動化の対象とした論理ＬＳＩは、大きさのほぼ揃った標準セルのみで構成されたレイアウトであった。その後、標準セルと、より大きいＲＡＭ、ＲＯＭ、ＰＬＡ等を混在して自動配置配線する要求が出てきた。このため、標準セルとマクロセルを混在する自動配置配線の手法の研究を始めた。レイアウト設計の階層に合わせて、先ず、ブロック内配置配線手法を開発した。他社からの発表は、いずれもマクロセルの位置を予め決定しておく方法であった。そこで、標準セルとマクロセルを共に自動配置配線する手法を開発した。次いで、ブロックを配置した後に使用するブロック間配線手法を研究した。ブロック間配線において、十年間近くも良い手法が発表されずに残っていた問題を実用的に解決する手法を開発した。これらの手法を組み込んだブロック内配置配線システム、ブロック間配線システムは、（株）日立製作所において論理ＬＳＩの製品に適用され、レイアウト設計の工数・期間を大幅に削減することに貢献した。その後も機能拡張を更に加えて、必要不可欠なシステムになっている。

１９９３年に岡山県立大学へ移った後は、京都大学上林教授の御指導の下でエキスパートシステムの知識であるルール表現の記憶最小化の研究を始めた。この研究は論理ＬＳＩのゲートアレー等で未配線が残った場合に配線入力するエキスパートシステムの大量のルールをコンパクトに記憶するための手法である。

また同時に、配線手法の基本問題の一つであるスタイナ木の研究を始めた。スタイナ木は配線長を最小にする配線パターンであり、従来、水平垂直配線方向に対応する直交幾何のスタイナ木が研究発表されていた。この配線方向の数を拡張して、任意のλ（正整数）個の方向を許すλ－幾何のスタイナ木の作成法を提案した。λ－幾何のスタイナ木は直交幾何のスタイナ木とユーグリッド幾何の間の隙間を埋めるものである。

目　　次

第1章　序論

1．1　本研究の目的

中小型計算機や周辺機器、端末機器など各種電子装置の小型化、低電力消費化、高信頼性化を図るために、論理ＬＳＩは不可欠なものになって来ている。これらの装置は生涯生産数が比較的少なく、搭載される論理ＬＳＩは多品種小量生産型であり、設計期間の短縮が最も重要になって来ている。また、製造プロセスの微細加工技術の進歩に伴い、チップに搭載されるトランジスタの数が数十万から百万個のオーダーに達して、論理ＬＳＩの論理規模が増加して来ている。論理ＬＳＩの論理規模が数万に達した１９８０年代に入って、レイアウト図面は巨大化し人手でレイアウトすることが限界になって来た。レイアウト設計を収束させるのが困難になり、設計期間が長くなることが目立って来た。

このため、計算機を用いて論理ＬＳＩの設計を支援するＣＡＤ、または、設計を自動化するＤＡが提案され、研究されて来た。その歴史は古く、１９６１年のプリント基板上の部品間の自動配線手法の報告[４１]にまでさかのぼることができる。当時より研究されてきた自動配置配線手法は、ＣＡＤ、ＤＡシステムに組み込まれ、１９８０年代の急激なニーズの高まりと共に実用レベルに達して来た。そして、自動レイアウトシステムが構築されて、設計工数削減期間短縮に大きく貢献している。

論理ＬＳＩのレイアウト設計は、セル、ブロック、チップからなる階層設計方式を採用している。この方式は、先ず、ＮＡＮＤ、ＮＯＲ等の基本ゲートを矩形の標準セルにして、標準セル列を積み重ねて配置配線しブロックを作成する。次いで、ブロックをチップ上に配置してブロック間を配線することによりチップをレイアウトを完成する。これまでのブロックはこのように標準セルのみで構成するか、または、人手作成された大きなＲＡＭ、ＲＯＭ、ＡＬＵ等が単独でブロックを構成していた。しかし、論理ＬＳＩの論理規模が増大するにつれて、ブロック内にＰＬＡを入れたいとの要求が出てきた。その理由は、チップが完成した後に発見された論理不良の修正をマスクパターン上で容易にできるようにするためである。本来、論理設計が完了してからレイアウト設計するべきであるが、ＬＳＩチップの完成期限に間に合わせるために論理設計がほぼ完了の段階でレイアウト設計に入ることが多い。それは、論理設計の最終段階で発見される論理不良は種々の特殊条件が重なった場合の論理不良で、発見するのに多大の時間を必要とするからである。このためＬＳＩが出来上がってから論理不良が発見されることがある。論理不良が発見されると、論理を修正検証して、再度レイアウトしなければならない。一般に、論理変更後の再自動レイアウト結果は、セルの配置位置が論理変更前と大きく変化してしまう。このため論理変更前のＬＳＩでの種々の特性検査結果は無駄になり、再検査の時間によってＬＳＩの完成を遅らせてしまう。ブロック内にＰＬＡを入れた場合、論理不良の修正がＰＬＡの結線変更のみで済むならば、セルの配置は変更はない。その結果、論理変更前のＬＳＩでの種々検査結果はほとんど無駄にならず、再設計期間を大幅に短縮できる利点がある。

ＰＬＡは標準セルと異なり側面に端子がありサイズも大きいので、従来のブロック内配置配線手法では、サイズの差異に起因する無駄領域が大きくてチップサイズを大きくしてしまうこと、および、側面の端子には配線できないという問題点があった。また一方で、ＲＡＭ、ＲＯＭ、ＡＬＵ等も混在するレイアウトの要求が出て来ることが予想された。これらは比較的簡単な規則的構造をしているので、人手でレイアウト設計することが容易であり、人手でレイアウトする方が集積密度が高くサイズを小さくできる。しかし、標準セルと比べるとサイズは数倍から数十倍も大きい。このため、ＲＡＭ、ＲＯＭ、ＡＬＵ等を、四辺に端子がある大きなマクロセルととして扱って、標準セルとマクロセルを混在できる配置配線システムの開発が開始された。

本研究の目的は、標準セルとマクロセルを混在できる配置配線手法を研究し、論理ＬＳＩのレイアウトシステムを開発することにある。特に、

（1）配線長最小問題の基本である最小スタイナ木の数学的側面を補強して、スタイナ木作成法を開発する。

（2）標準セルとマクロセルのサイズの差異を整合させたレイアウトモデルを定義して、このレイアウトモデルに合った配置配線手法を開発する。

ことをを目的としている。これらの自動配置配線手法を組み込んだ論理ＬＳＩのレイアウトシステムは現在実用に供されている。

1．2　本研究の位置付け

計算機によるレイアウト設計の自動化の研究は１９７０年代より始まっていたが[６２][３４][１２][３]、１９８０年代に入ると実際の製品の論理ＬＳＩに適用され始め、発表も多くなって来た[９][５][１４][２７]。１９８０年代半ばまでの論理ＬＳＩは、標準セル列を積み重ねた形のレイアウトがほとんどであった。しかし、論理ＬＳＩの論理規模が増大するにつれて、論理ＬＳＩの中にＲＡＭ、ＲＯＭ、ＰＬＡ等を取り入れる要求が出てきた。従来の標準セルを並べる自動配置配線技術では、サイズの差異に起因する無駄領域が大きくてチップサイズを大きくするため、一緒に扱うことは出来なかった。このため、ＲＡＭ、ＲＯＭ、ＰＬＡ等のマクロセルは単独でブロックとしてチップ内の予め決めた位置に配置し、標準セルのみ別に自動配置配線したブロックにして配置して、両者の間を配線するビルディングブロック方式の設計が採用されていた。

論理ＬＳＩが多品種設計されるに伴い、ＲＡＭ、ＲＯＭ、ＰＬＡ等のマクロセルはその論理ＬＳＩに合致したものが使用されるようになり種類が増加してきた。大小さまざまなサイズのマクロセルを論理ＬＳＩに搭載する必要が出てきた。一方、論理ＬＳＩの設計期間短縮は至上命令であるので、さまざまなサイズのマクロセルと標準セルを混在して自動配置配線する必要が出て来た。比較的小さなマクロセルと標準セルを混在させて自動配置配線する方法は発表されていたが[５３]、標準セルを配置する領域とマクロセルを配置する領域を予め設定しておいて、標準セルのみ自動配置し、全体を自動配線する方法であった。これに対して、セルの配置位置を予め設定することなく、標準セルとマクロセルを同時に自動配置配線する方法を開発した[７０]。本方法は実用化され、実際の製品の論理ＬＳＩに適用されている。

ビルディングブロック方式の設計では、比較的大きなマクロセルは単独のブロックとし、標準セルを配置配線して作成したブロックと共にチップ上に配置する。ブロック間

を自動配線してレイアウトする。このブロック間配線に高速なチャネル配線法を適用する場合、分割領域であるチャネルの接合関係を表わすチャネル順序グラフにサイクルがあると配線不能であるという問題が１９７３年より知られていた[３３]。この問題は難しく十年近くも良い方法が発表されることがなかった。発表されていた方法は、ブロックを移動してチャネル構造を変更し、チャネル順序グラフにサイクルが現われないようにする方法[３５]であった。しかし、ブロックを移動してサイクルを解消すると、ブロックの移動に伴い無駄な領域が発生してチップサイズを大きくしてしまう問題があった。実際の論理ＬＳＩのレイアウト結果を調べてみるとサイクルが現われることは少なく、現われてもチャネル４個からなる単純なサイクルがほとんどであった。複数のサイクルが錯綜してできる複合サイクルが現われるのは稀であった。このため、単純サイクルが現われた場合に、サイクル内の４個のチャネルを巡回しながら配線して配線を収束させる方法を開発した[７８]。同時期に、チャネル２個をＬ字形に切り出して配線することによりサイクルを切断する方法が発表された[１１]。このＬ字形チャネルを切り出す方法は、これまでに発表された方法の中で最も良い方法として現在定着している。しかし、現実に現われるサイクルはほとんど単純サイクルであるので、開発した方法は実用上問題なく、実際の製品の論理ＬＳＩに適用されている。

現在もっとも普通に使用されているチャネル配線法は、配線領域をチャネルと呼ぶ矩形領域に分割して、チャネルごとに配線した後全体をまとめる方法である。このため、チャネル配線法を適用する前に各信号の配線経路をどのチャネルを経由させるかをきめるグローバル配線処理が必要である。グローバル配線において、各信号の最短配線はスタイナ木の形である。従来、スタイナ木は垂直水平の直交配線に対応する直交幾何とユーグリッド幾何の分野で多くの発表がされている[２１][２８][２９][５９]。しかし、論理ＬＳＩは、製造プロセスの進歩により多層配線が可能になり、５～６層配線が現実になって来た。また、論理ＬＳＩの高速化が進みより短い配線が必要になって来た。このため、多層プリント基板に採用されている斜め方向の配線がいずれ採用されると思う。この垂直水平斜め方向配線に対応するスタイナ木の作成法を考案し、更に配線方向を一般化したλ－幾何のスタイナ木作成法を提案した[７３]。λ－幾何のスタイナ木に関する発表は未だ少なく[４][５１]、今後発展する分野である。

１．３　本研究の構成

第２章では、従来の論理ＬＳＩ設計の概要と代表的な自動配置配線手法を述べる。第３章、第４章で、配線長最小問題の基本であるスタイナ木の作成法を述べる。第５章、第６章で、標準セルとマクロセルを混在した論理ＬＳＩの階層設計に使用される配置配線手法を述べる。第７章ではゲートアレーの未配線等を扱うエキスパートシステムの大量の知識を効率良く記憶する手法を述べる。第８章で本研究の結論と今後の研究課題を述べる。

第２章では、従来の論理ＬＳＩ設計の概要とその中の階層的レイアウト設計方式について述べる。自動配置処理は、普通、初期配置手法と配置改善手法の２段階で構成される。この各々で使用される代表的な手法を述べる。自動配線手法は、汎用型とモデル限定型に分類できる。この各々で使用される代表的な手法を述べ、特に、最も普通に使用されるチャネル配線法について詳述する。チャネル配線法は第５章、第６章で使用する。

第３章では、配線長最小問題の基本であるスパンニング木とスタイナ木の性質と両者の関係について述べる。更に、従来の垂直水平２方向を拡張して、配線方向の数を任意のλ（正整数）個許すλ－幾何の説明を述べる。λ－幾何は第４章で使用する。

第４章では、λ－幾何（$\lambda \equiv 1, 2 \pmod 3$）のスタイナ木の性質と新たな作成法を提案する。先ず、λ－幾何の距離の計算式を導き出す。そして、３点の最小スタイナ木のスタイナ点の位置の予想を提出し、３点が特別な位置関係にある場合にこの予想が正しいことを証明する。次に、この予想を用いて、３点及び４点のスタイナ木の幾何学的構成法を提案する。そして、与えられた任意個数の点に対する多点スタイナ木の作成法を提案する。この方法は、λ－幾何の最小スパンニング木を出発点として、３点／４点の部分木を３点／４点のスタイナ木で置き換えることを繰り返す方法である。そして、計算機を用いた実験をして、最小スパンニング木から３.２％短縮できることを述べて本生成法の有効性を示す。また、従来の垂直水平配線に斜め方向の配線を追加することが配線長を１０％程短縮する大きな効果があることも示す。

第５章では、標準セルとマクロセルを混在できるレイアウトモデルと配置配線手法を述べる。ブロックは、階層レイアウト設計の下位階層レイアウトである。従来の自動レイアウトの対象となる標準セルはサイズ、形共に大きな差異はない。一方、ＲＯＭ、ＲＡＭ、ＰＬＡ、ＡＬＵなどのマクロセルは、チップの実装密度を大きくするのに有効であるが、標準セルと比べてサイスが数倍から数十倍大きい。先ず、この２種類のセルのサイズの差異による無駄領域を削減する拡張レイアウトモデルを提案する。拡張レイアウトモデルでは、比較的小さいマクロセルは標準セル列の中に入れ、大きいマクロセルは単独で標準セル列と同格にして高さの大きいセル列の一つとして扱う。次に配置配線手法を述べる。大きいマクロセルの幅に合わせてブロックをサブブロックに分割し、サブブロック内は従来の手法でセルを配置する方法を述べる。マクロセルは標準セルと異なり左右側面の端子にも配線する必要がある。チャネル配線法を用いてこの側面端子の引き出しに必要な配線領域を小さくする配線方法について述べる。

第６章では、ブロック間配線の方法について述べる。ビルディングブロック方式でのブロックの配置は、直線によって再帰的に２分できるスライス構造配置と再帰的には２分できない非スライス構造配置に分かれる。非スライス構造配置をチャネル順序グラフで表現した時に現われるサイクルを単純サイクルと複合サイクルに分類して、複合サイクルはブロックを移動して単純サイクルにしておく。単純サイクルの非スライス構造配置を配線する「単純サイクル巡回配線法」を提案する。本方法は、チャネルグラフ上の４つの頂点からなるサイクルに対応するチャネルを巡回して、チャネルの接合部の座標

値が一致するまで繰り返し配線する。何回も繰り返し配線しても一致しない時は、3辺固定チャネル配線を用いる。実験の結果は繰り返し配線のみで十分であった。

第7章では、ゲートアレーの未配線等を扱うルールベースのエキスパートシステムの大量の知識を効率良く記憶する手法について述べる。エキスパートシステムのプロダクションルールは論理関数で表わすことができる。ルールベースを論理関数の積和形で表現し、二分決定グラフを用いて積和形を高速に最小化することによって、ルールベースのルール数を削減する。積和形表現から等価な積項を等価クラスとして選び出した後、等価クラスに基づいて積項数が最小になるように含意関係（ルール）を書き換える。特に、論理関数の含意関係をグラフ表現した時の強連結成分をサイクルに書き換えることによって、表現コストを大幅に削減する「巡回含意法」を提案する。そして、実験により有効性を確かめる。

また、論理関数の積項の等価クラスを表現するのに、関係データベースの等価キーによる表現を応用できることを示す。この表現は積和形による表現よりも記憶コストが小さくできることも述べる。

第2章　論理ＬＳＩの自動設計

2．1　はじめに

本章では、先ず論理ＬＳＩの設計の概要を述べる。続いて、論理ＬＳＩの自動レイアウト設計に用いられる基本的な自動配置配線手法を述べる。配置手法は初期配置手法と配置改善手法に分類され、通常は両者を組合せて使用される。レイアウトの最大の目的はチップサイズの縮小である。配置処理は、配線すべき端子の位置を決め配線量を決めるので、配線問題の難易を大きく左右し、重要である。配線手法は汎用型とモデル限定型に分類される。後者は種々のモデルが提案されているが、最も普通に使用され、第5章、第6章で使用するチャネル配線法を特に詳述する。

2．2　論理ＬＳＩの設計工程

論理ＬＳＩの設計工程を図2．1に示す。設計工程は、大きく方式設計、論理設計、レイアウト設計、に分かれる。方式設計では論理ＬＳＩの機能性能の仕様を決め、論理ＬＳＩの機能を実現するシステム構成を決定する。

論理設計では、先ず、機能論理設計において論理ＬＳＩ全体を論理機能ブロックに分割する。論理機能ブロックとは、演算ブロック、マイクロプログラムＲＯＭ、デコードブロック、キャシュブロック、入出力制御ブロック等である。論理機能ブロックの機能を決定して、これらのインターフェース信号を決める。そして、機能論理レベルで論理機能を検証する

次いで、各論理機能ブロック内の詳細論理設計をする。機能論理設計の結果に基づいて、論理機能ブロックをマルチプレクサ、フィリップフロップ、ゲートのレベルの論理要素まで詳細化して、これらの結線情報を作成する。作成した詳細論理は、詳細論理シミュレータを用いて検証する。論理に不良があれば論理を修正して論理検証することを繰り返す。同時に、ＬＳＩ製造後の診断を目的として診断可能性も調べる。故障の診断の発見の割合が低い場合は診断論理の修正変更をする。そして、診断データを生成してチップ製造後に各チップが正しく機能を実現しているかどうかの検証に備える。

回路設計では、フィリップフロップ、ゲート等の論理要素をトランジスタレベルの回路で構成する。また、比較的規則性があり人手設計の方が実装密度が上がるＲＯＭ、ＲＡＭ、ＡＬＵ等もトランジスタレベルの回路で構成する。そして、回路シミュレータにより回路検証する。

詳細論理が完成したらレイアウト設計する。レイアウト設計は、詳細論理と論理で用いた部品（セル）の幾何情報を入力として、チップのレイアウトパターンを出力する。レイアウト設計ではチップ面積と遅延時間などの電気特性を最適化する。レイアウト設計は、次に述べるセル、ブロック、チップの階層設計方式で行う。レイアウトパターンは部品内部のパターンを組み込んで、ＬＳＩのマスクデータに変換する。

以上の設計が終わると、マスクを作成して、製造ラインにて論理ＬＳＩのチップを製造する。チップはパッケージに組み立ててから出荷検査する。

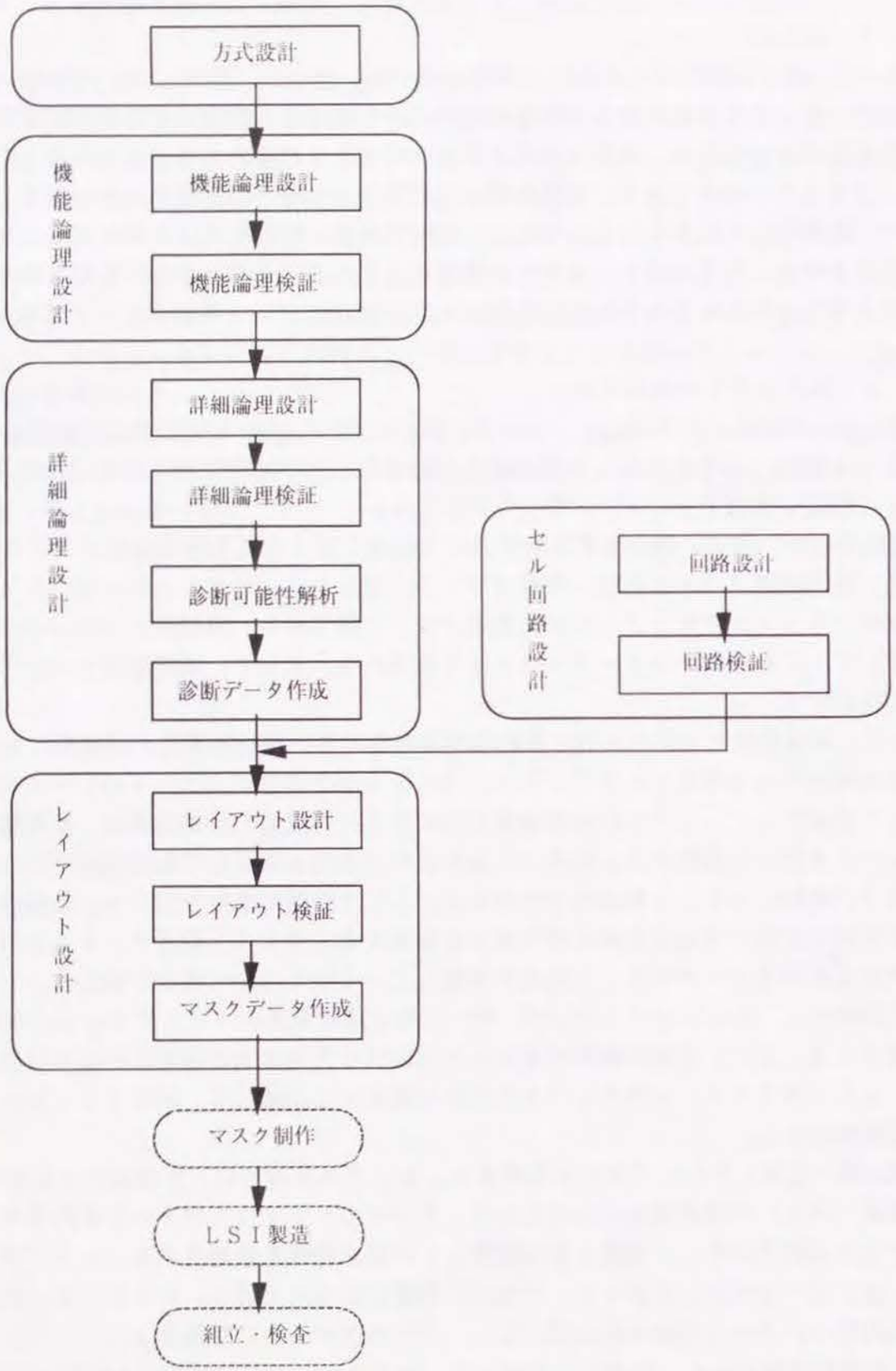


図２．１　論理ＬＳＩ設計工程

２．３　階層レイアウト設計方式

論理ＬＳＩの論理設計、レイアウト設計は階層設計方式を採用している。目的は、論理のサイズを論理設計者が扱易い規模に細分化すること、複数の設計者が同時並行設計できること、及び、細分化された論理ごとに纏まった配置配線することによって論理のディレイを管理し易くすることである。具体的には、図２．２に示すように、チップをブロックに分割し、更にブロックはセルに分割する。セルは、フィリップフロップ、ゲート等の論理要素であり、数個から数十個のトランジスタから構成されるレイアウトの最小単位である。ブロックは、機能の纏まった論理で設計者が扱易い数百ゲートの規模の論理で構成され、セルを配置配線してレイアウトされる。

レイアウト設計では先ずチップ全体を見たトップダウン設計をした後、最小構成要素であるセルを配置配線してブロックにし、次いでチップに構成していくボトムアップ設計をする。この各段階に対応する自動設計用ツールが開発されている。

次に示すレイアウト設計の流れの（１）（２）がトップダウン設計であり、（３）～（５）がボトムアップ設計である。

（１）レイアウト階層論理の構成

論理設計終了後、論理をレイアウト用にチップ、ブロック、セルの３階層に纏める。論理の各要素にセルを割付け、機能的な塊として数百ゲート規模の論理をブロックとして纏める。

（２）フロアプラン設計

チップ全体のイメージをつかむためにブロックの形状と相対位置を決める。先ず、ブロックを構成するセル列数をパラメータとしてブロック内配置配線を実行し、各ブロックのアスペクト比が変化した形状データを作成する。ブロックの形状データの中から適当なアスペクト比の形状を選択して、ブロックの相対配置をする。ブロック配置は、ブロック相互のデータや制御信号の流れやＬＳＩピンを考慮して相対位置関係を決める。

（３）ブロック内配置配線設計

フロアプランで決めたセル列数、外部端子の引き出し方向、特定セルの配置位置などの制約条件に従って、各ブロックのレイアウトを行う。この結果ブロックの最終形状、外部端子の位置が確定する。

（４）ブロックの配置

レイアウトが完了しサイズが確定したブロックをチップ上に配置し、配置の微調整をする。

（５）ブロック間配線設計

配置された各ブロックの外部端子間を配線する。

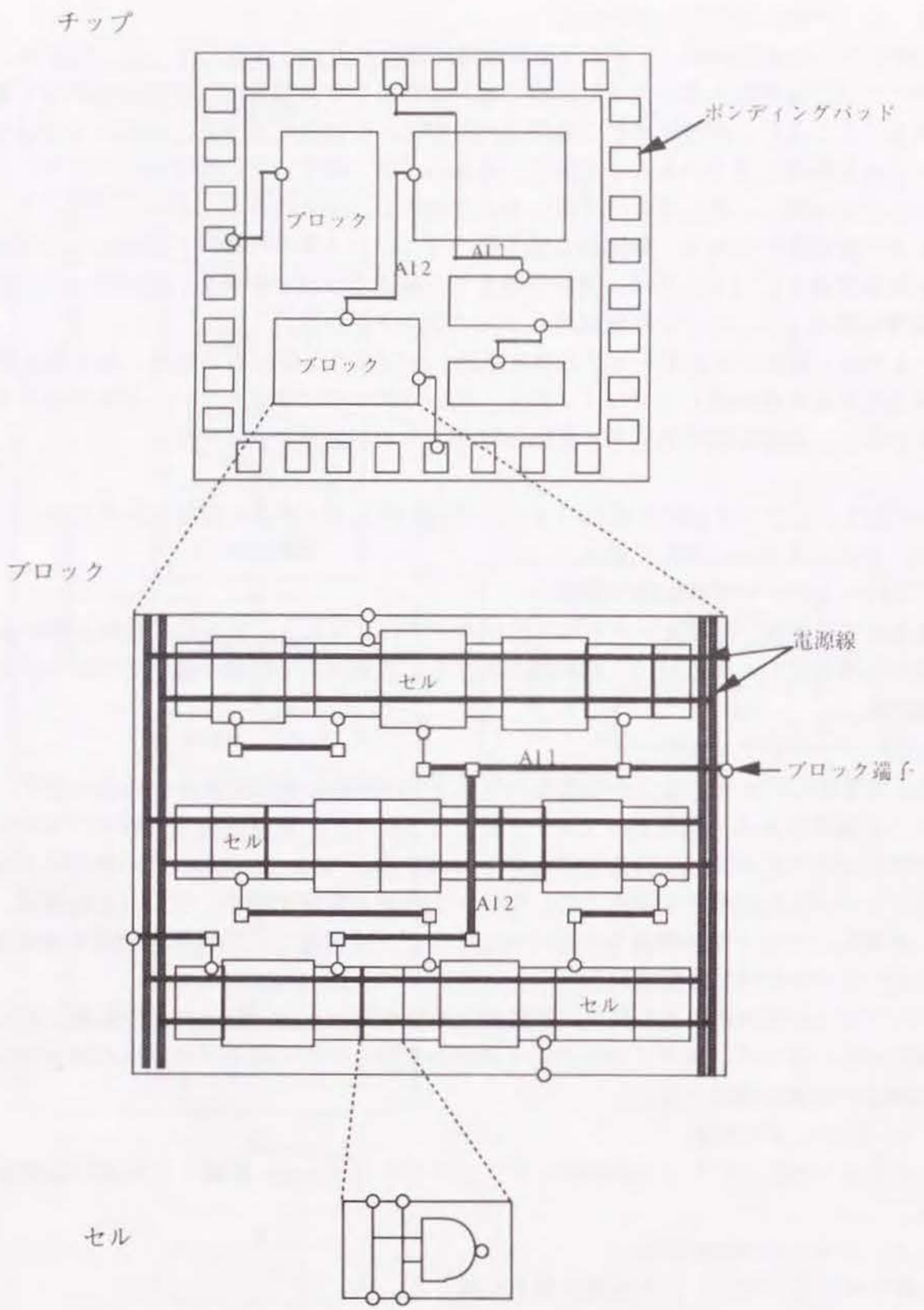


図２．２　階層レイアウト設計方式

２．４　自動配置手法

本節では、配置問題を定義し、自動配置手法について説明する。

配置問題：論理ＬＳＩのレイアウトにおける配置問題とは、以下の制約条件下で目的関数を最適化する組合せ問題である。

（１）配置の単位（ブロック、セル）は、大きさ固定の矩形である。

（２）配置単位の各矩形の間には配線の数が与えられている。

目的関数は、配線領域を含めたレイアウトの対象（チップ、ブロック）の面積の最小化である。

この配置問題を解くには、目的関数の配線領域面積を求める必要があり、このままでは解くことができない。正確には配線処理をしないと配線領域面積は求まらない。１つの配置ごとに配線処理することは実際上不可能である。このため、目的関数として採用される代表的なものに次のものがある。

（１）配置の段階で仮想的に決定した配線経路長の総和（仮想配線長合計）。

（２）配置対象領域を仮に分割する線と交差する配線数（カット数）。

（３）配線領域を仮に分割して、各分割領域を通過する通過する配線数（混雑度）。

配置問題は、普通、構成的配置手法による初期配置処理と漸近的配置改善手法による配置改善処理との２段階の処理で解く。配置問題を解くためのヒューリスティックな手法は数多く提案されている。

２．４．１　初期配置手法

初期配置手法として代表的な方法を述べる。

（１）　クラスタリング１次元配置法[５６]（図２．３）

配置要素間に結合度を定義して、結合度が大きい配置要素をまとめてクラスタとする。更に、結合度が大きいクラスタどうしをまとめてクラスタを作成し、最終的に２分岐クラスタ木を作成する。ここで、結合度とは、各配置要素につながる配線本数に対する、２つの配置要素またはクラスタ間の配線本数の割合である。クラスタ木ができたら、クラスタ木の根から葉に向かって、２分岐に従う２分割を繰り返す。２分割するごとに、分割して得られた子クラスタまたは配置要素の左右相対配置を、仮想配線長が小さくなるように決める。こうして１次元配置を得る。

（２）ミニカット配置法[３]（図２．４）

配線領域を水平または垂直に切断する線をカットラインとする。カットラインの両側に分かれた配置要素に関する目的関数を最小化する手法である。目的関数としては、信号数が用いられる。

先ず、カットラインの処理順序を決める。そして、カットラインを選択する。カットラインにより切断される配置要素の部分集合の中で、カットライン上の信号数が最小となるように配置要素を入れ換えしながら更に２つの部分集合に分ける。これを、処理するカットラインがなくなるか、分割後の部分集合が１つの配置要素を含むまで繰り返す。

（３）　クラスタリング２次元配置法[３６][４０]（図２．５）

クラスタリング１次元配置法と同様に２分岐クラスタ木を作成する。クラスタ木ができたら、同様に、クラスタ木の２分岐に従う２分割を繰り返す。配置領域もカットラインによる２分割を繰り返しておく。クラスタ木を２分割するごとに、配置領域の垂直カットラインの左右、または、水平カットラインの上下に子クラスタを配置する。左右上下の位置は仮想配線長が小さくなるように決める。こうして２次元配置を得る。

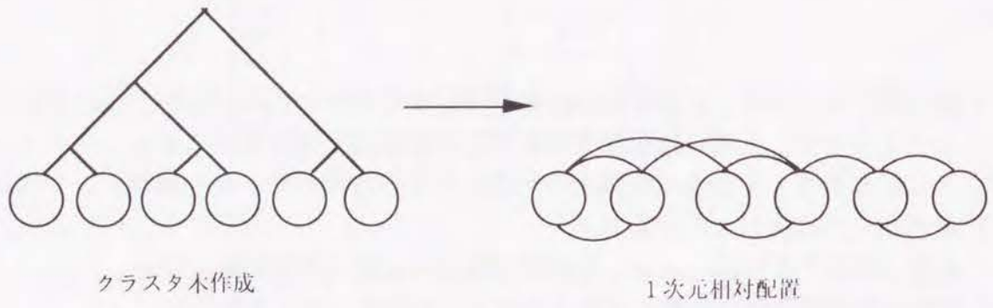


図２．３　クラスタリング1次元配置法

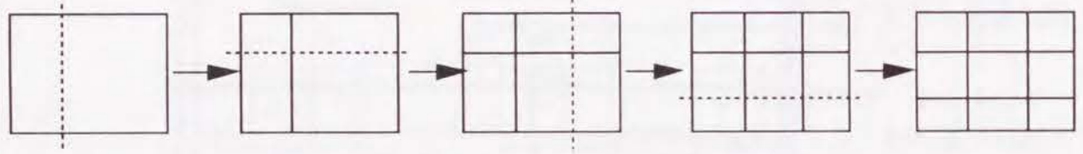


図２．４　ミニカット配置法

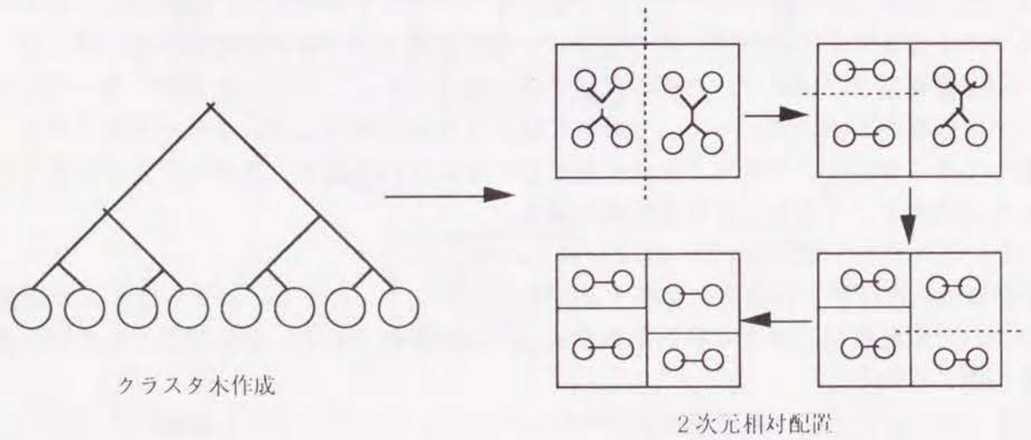


図２．５　クラスタリング２次元配置法

２．４．２　繰り返し配置改善手法

（１）スタインバーグ法

共通の接続がない配置要素の部分集合を見つけて、この配置要素を配置から一旦取り除いて空場所をつくる。次に、取り除いた配置要素を仮想配線長が短くなるように再配置する。再配置は１次元割付問題として定式化して解く。

（２）ペア交換法

２個の配置要素の位置を交換し、新しい配置の配線長を計算する。もし改善があれば新しい配置に置き換え、改善がなければそのままにして置き換えないことを繰り返す配置法である。

（３）緩和法

配置要素間の配線本数を張力の大きさとし、つながった配置要素から受ける張力の働く方向の合成ベクトルを求める。この合成ベクトルの方向に配置要素を移動することを繰り返して、力学的に安定な位置に配置する。即ち各配置要素について、それにつながる配置要素から求まる合成ベクトルが小さくなるように配置を改善する。

（４）ネットバランス法[６７]

１次元配置の中で配置改善する方法である。配置要素に関して右方向に広がるネット（信号）と左方向に広がるネットの数を等しく（バランス）する位置が配線領域内の配線長総和を削減するという考え方に基づいて繰り返し配置改善する。

（５）シミュレーティッドアニーリング法

本手法は、比較的単純なペア交換法等を繰り返す際に、改善がある場合のみでなく改善がない場合もある確率で採用するメタ方法である。この確率を制御する方法が特徴で、アニール（焼きなまし）の如く初めは大きい確率とし、徐々に確率を小さくしていく。従来の配置改善法は、新しい配置の目的関数がよくなれば配置改善し、よくなければ配置を変更しないという考え方に基づいていた。この考え方では必ずしも大域的最適解に到達できず局所最適解に陥る可能性が大きかった。そこで、本手法は、確率的に目的関数がよくない場合も採用することにより、局所最適解に陥る可能性を小さくする。

２．５　自動配線手法

本節では、配線問題を定義し、自動配線手法について説明する。

従来の配線手法は、配線格子を設定し、配線格子の上で配線する。この配線格子を図２．６に示す。配線格子とは、配線が隣接して通過しても設計ルールに違反しない最小間隔を基に作成する。通常縦横方向共にスルーホールと配線パターンが隣接できる場合のスルーホール中心と配線パターンの中心線間隔を配線格子間隔とする。

従来の配線手法は以下を仮定する。

（１）前述の配線格子

（２）配線層は２層で、縦方向と横方向は異なる配線層を使用する。

（３）端子は配線格子の１格子点とする。

（４）配線の幅は１種類とする。

これらは配線の問題の定式化を容易にするために置いたモデルの制限条件である。

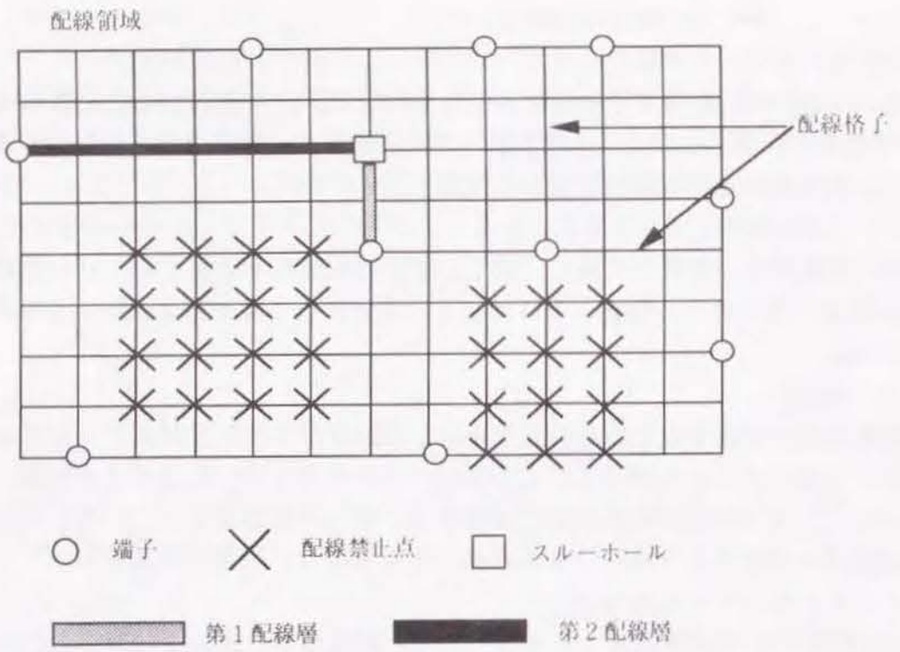


図２．６　配線格子モデル

配線問題：論理ＬＳＩのレイアウトにおける配線問題とは、以下の制約条件下で目的関数を最適化する組合せ問題である。

（１）配線モデルは、上に述べたものに従う。

（２）信号ごとに配線すべき端子の集合が与えられている。

目的関数は、配線領域面積の最小化である。

配線手法は、大きくグローバル配線手法と詳細配線手法の２つに分類できる。配線手法の分類を図２．７に示す。歴史的には、プリント基板を対象にして、汎用型の迷路法、と線分探索法が開発され、次いで、モデル限定型の詳細配線法である、チャネル配線法やスイッチボックス配線法やリバー配線法が開発された。モデル限定型の配線手法は、配線領域を小規模な配線領域に分割して適用する。このため、モデル限定型の配線手法を適用する場合は、次の３段階の処理で構成する。

（１）配線領域をモデルに合う小規模配線領域に分割する。

（２）グローバル（概略）配線法を用いて、どの分割領域を通過させるかを決める。この段階では、まだ詳細な配線パターンは決めない。

（３）モデル限定型の配線手法を用いて、各分割領域内の詳細な配線パターンを決める。

大規模の論理ＬＳＩは、配線領域を小規模な配線領域に分割して、モデル限定型の配線手法を適用することが多い。

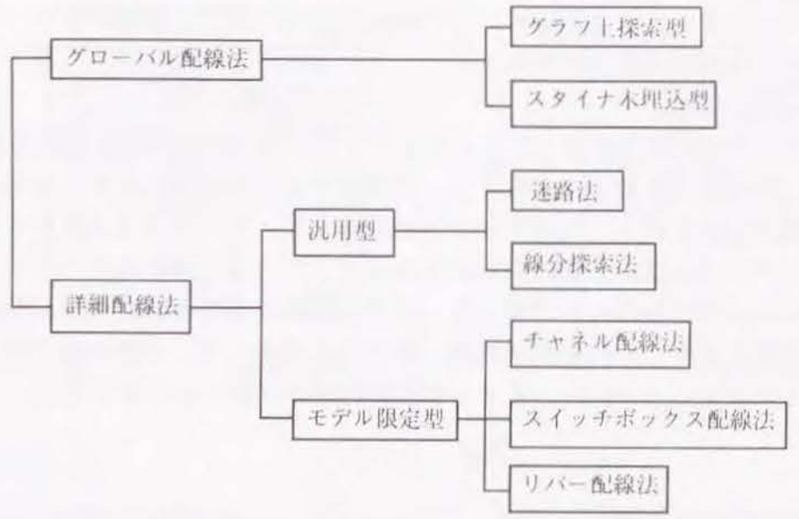


図２．７　配線手法の分類

２．５．１　グローバル配線手法

グローバル配線手法は、モデル制限型の詳細配線手法を適用する前に用いられる。論理ＬＳＩでは、多数の信号を最短経路で配線しようとすると、中央部に配線経路が集中して配線が困難になる。配線を分散させて配線密度を平準化する必要がある。このため、論理ＬＳＩの配線領域を小規模な領域に分割した後に、グローバル配線手法は、各信号をどの分割領域を通過させるか決める。また、分割領域間のインタフェース情報を決定する。グローバル配線手法は、普通、分割配線領域の接合部を点として、配線領域の配線方向に枝をつけたグラフの上で経路を探索する。各枝には配線が通過できる本数を枝の容量として設定し、この枝の容量を超えない範囲で配線経路を決定する。セル列を積み重ねた形のゲートアレーやビルディングブロック方式のブロック内では、セル列の中央を通る線とセル列に直交する線で配線領域を格子状に分割してできる、格子グラフ（図２．８(a)）を用いる。また、ビルディングブロック方式のブロック間配線には、ブロック隙間の配線領域を矩形の集合に分割してできる、チャネルグラフ（図２．８(b)）を用いる。

主なグローバル配線手法には、グラフ探索型とスタイナ木埋込型がある。

（１）グラフ探索型

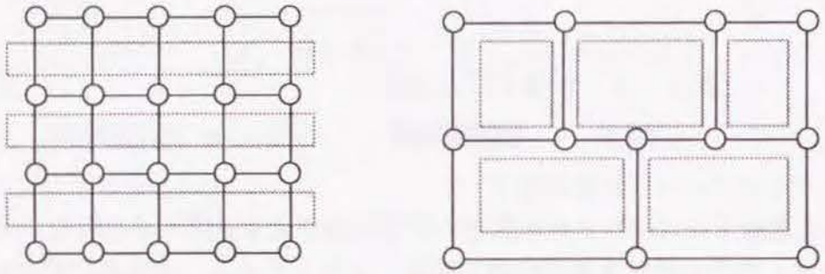
格子グラフまたはチャネルグラフに、配線する信号の２端子の位置に２点を追加して、２点間ををつなぐ配線経路をグラフ上で探索する。グラフ探索型で２端子間の経路を決めていくことは、多端子の信号ではスパンニング木を作ることに対応する。代表的な手法はダイクストラの最短経路アルゴリズムである。最近は多品種ネットワークフローの手法を用いるものもある。

枝の容量を超えて配線経路を割り付けることはできない。このため、経路の混雑の関数を設けて、枝の容量を超えないように配線経路を制御する。また、枝の容量を無視し

て配線経路を割り付けた後に、容量を超えた枝の配線経路の一部を引き剥がして、余裕のある枝に迂回させて再割付する方法もある。

（２）スタイナ木埋込型

多端子の信号では、スパンニング木よりもスタイナ木を作成した方が配線経路の長さがより短くなる。このため、多端子の信号では、垂直水平方向の直交スタイナ木が用いられる。各信号の端子位置を点としたスタイナ木を作成して、格子グラフまたはチャネルグラフと重ねる。グラフの枝の容量に余裕があればスタイナ木を埋込む。グラフの枝の容量に余裕がなければ迂回経路にして埋込む。また、枝の容量を無視して配線経路を割り付けた後に、容量を超えた枝の配線経路の一部を引き剥がして、余裕のある枝に迂回させて再割付する方法もある[４３]。スタイナ木の作成法は第４章で詳述する。



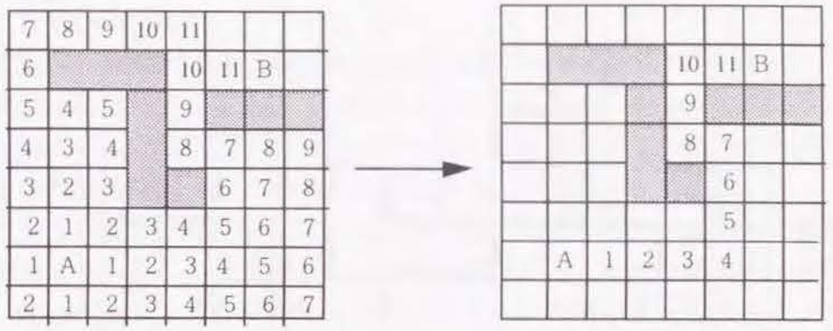
（ａ）格子グラフ　　（ｂ）チャネルグラフ

図２．８　グローバル配線手法が用いるグラフ

２．５．２　詳細配線手法

（１）　迷路法

本手法は、Lee[４１]により開発された歴史的に最も古い汎用型の配線法である。基本的な考え方は、図２．９に示すように、配線領域全体の配線格子点のマップをメモリ上に作成し、始点端子から波面が水面を広がるように隣接格子点に広げて行き、同時に始点からの距離を記憶していく。波面が目標端子に到達したら、今度は波面を逆にたどる。目標端子から距離が１ずつ少ない隣接点をたどって、始点端子までの配線経路を選択する。１つの配線経路を完成したら、完成した配線パターンを配線済み固定パターンとして、次の配線に対して配線禁止扱いする。この処理を各配線ごとに繰り返す。この方法は迷路法という名前が示すように、配線経路が存在すれば必ず最短経路を発見できる特徴がある。しかし、配線結果が配線の順序に大きく依存すること、経路探索の処理時間と所要メモリが膨大であるという問題点がある。

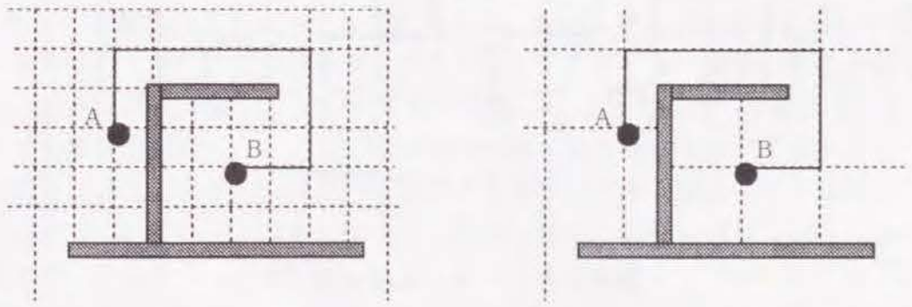


（ａ）ＡからＢへの波面の展開　　（ｂ）ＢからＡへの逆探索

図２．９　迷路法

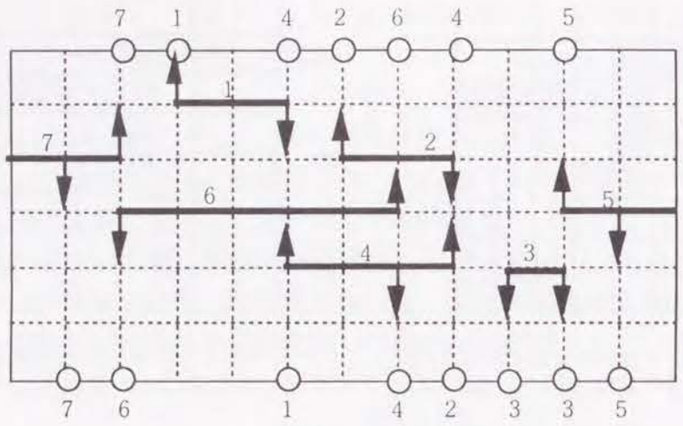
（２）線分探索法

本手法は、三上、田淵[４６]とHightower[２３]により、迷路法の処理時間、所要メモリ量の問題を解決するために独立に開発された汎用型の配線法である。。基本的な考え方は、始点端子から配線格子に従って直線を障害物に衝突するまで発生し、直線が障害物に衝突したら別の方向に線分を発生させることを繰り返して、目標端子に到達したら、今度はこの線分を逆にたどる。目標端子から線分の結合関係をたどって始点端子までの配線経路を選択する。一つの配線を完成したら、完成した配線を障害物として次の配線処理をする。三上、田淵の方法とHightowerの方法は、図２．１０に示すように、線分を発生させる点の位置が異なる。前者の方法は、直線上のすべての格子点で直交する直線を発生させる。このため、配線経路が存在すれば必ず経路を発見できる特徴がある。しかし、最短経路を見いだすとは限らない。後者の方法は、直線上の１点でのみ直交する直線を発生させる。このため、配線経路が存在しても発見できない場合がある。。この方法は配線格子点のマップを作成する必要がなく、迷路法よりも高速で、所要メモリ量も少なくて済む。本方法は比較的折れ曲がりの少ない配線パターンを作成する。しかし、配線結果が配線の順序に大きく依存する問題点が残る。

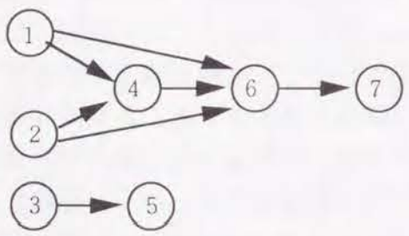


（ａ）三上、田淵の方法　　（ｂ）Hightowerの方法

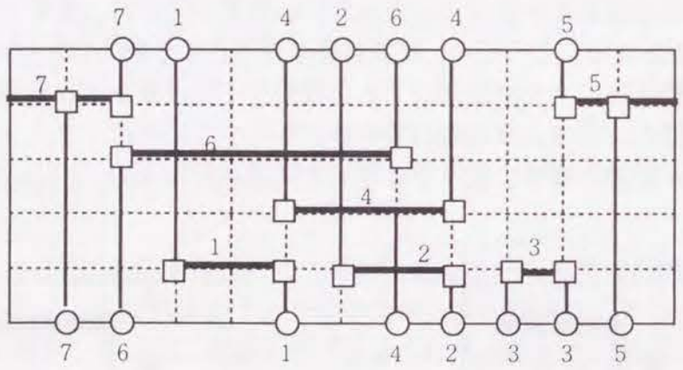
図２．１０　線分探索法



（ａ）与えられたチャネル配線問題



（ｂ）制約グラフ



（ｃ）チャネル配線結果

図２．１１　チャネル配線法

（３）チャネル配線法

本手法は、Hashimoto & Stevens [２２] により開発された、配線領域をチャネルと呼ぶ矩形領域に分割してチャネルごとに配線する手法である。特徴は配線結果が配線順序に依存しないこと、分割処理なので高速で所要メモリ量が少ないことである。この特徴と配線問題の大規模化に伴い、本手法は現在最も広く使用され、論理ＬＳＩの配線はほとんどがチャネル配線法を使用している。迷路法や線分探索法が適用前に配線領域や端子位置が固定していることを前提とし、配線領域が不足すると配線不能（未配線）を発生するのに対して、チャネル配線法はチャネル上下辺の端子を平行移動することによって配線領域を必要最小限確保するので、１００％配線し未配線を発生させない。

チャネル配線問題を。図２．１１に示す。チャネルとは、矩形の領域で上下辺にＸ座標が固定した端子が並んでいる。端子のＹ座標と矩形の左右辺からチャネル外へ出る配線の切り口のＹ座標は浮動で、配線が完了した時点で左右辺の長さ（チャネル幅と呼ぶ）を必要最小限にすることよって決まる。チャネル配線法の目的関数はチャネル幅の最小化である。チャネル配線は、配線を横方向の幹線と縦方向の支線で実現し、幹線と支線を異なる配線層に割り当てる。従って、幹線と支線の交点には必ず２つの配線層を結ぶスルーホールが必要である。図２．１１(a) では、幹線を実線で支線の上下方向を矢印で示す。配線経路は、幹線を横方向の配線格子（トラックと呼ぶ）上に割り当て、幹線から端子まで垂直に支線を引く。

チャネル配線問題の解法は多く提案されている[３４][１２][５７][５]。実用的なチャネル配線法[６２]を述べる。幹線の集合において垂直制約関係と水平制約関係を表す制約グラフを導入する。垂直制約関係は、支線が重なって短絡するのを回避するために幹線の上下関係を規定する。水平制約関係は幹線のＸ方向区間が重なるために同一トラック上に配線不可能であるという制約を規定する。制約グラフは、幹線を頂点とし、２種類の枝をつけることにより作成する。２つの幹線が垂直制約関係にある時、下側の幹線から上側の幹線に向かう有向枝をつける。また、２つの幹線が水平制約関係にある時、両幹線間に無向枝をつける。垂直制約関係がない場合は、束論の Dilworth の定理により最小チャネル幅（幹線の区間の重なり数の最大値）で配線することができる[２２]。特に垂直制約関係のみを表わしたグラフを垂直制約グラフと言う。

通常、チャネル配線はトラックごとに配線する。先ず、垂直制約関係にない幹線の部分集合を取り出し、更に、この部分集合内で水平制約関係のない幹線の集合を選んで１トラックに割り付ける。図２．１１(b) では、下側のトラックに対して幹線１、２、３からなる部分集合を選択し、これらが水平制約関係にあるかどうかをチェックして、トラックに割り付ける。水平制約関係があれば、水平関係がある幹線のどちらか一方はトラックに割当ない。制約グラフからトラックに割り付けた幹線の頂点と頂点につながる枝を除去して、次のトラックの配線に移る。これを繰り返す。

制約グラフに垂直制約関係の有向枝によるサイクルがある場合には、すべての幹線を直線のまま配線することは不可能である。サイクルがある場合にはサイクル内の頂点に対応する幹線を分割し屈曲した配線をする必要がある（図２．１２）。

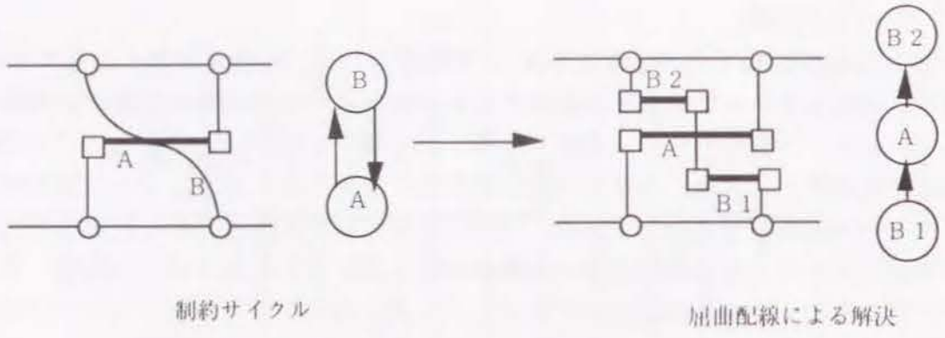


図２．１２　制約サイクルの解決

制約グラフに垂直制約関係の連鎖、即ち、有向枝による長いパスがある場合には、幹線密度（幹線の区間の重なり数）を大幅に超過したトラック数を必要とする。このため、パス内の頂点に対応する幹線を分割し屈曲した配線をする必要がある（図２．１３）。

これらの幹線を分割し屈曲した配線をする場合、難しいのはどの幹線をどの位置で分割するかという問題である[１]。幹線を分割すると分割幹線間をつなぐ支線が必要である。分割位置のＸ座標に他の信号の端子があれば、この信号の端子と幹線をつなぐ支線と分割幹線間をつなぐ支線が短絡してはならない。即ち、この信号の幹線と分割幹線には垂直制約関係が発生する。

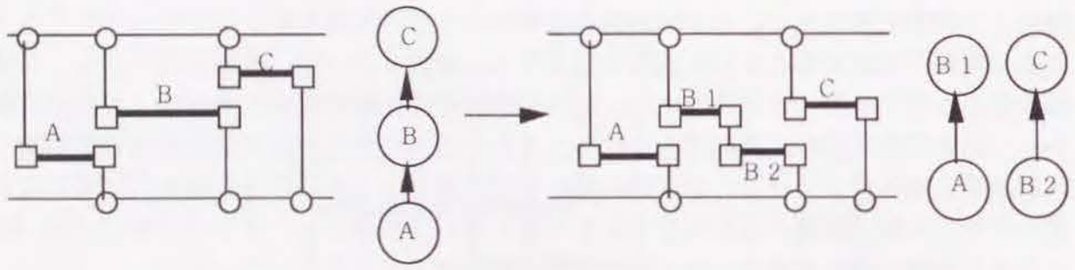


図２．１３　制約パスの短縮

複数のチャネルがＴ字形に接合している場合には、チャネルの配線順序に制約がある。配線領域をチャネルに分割した時、図２．１４のように２つのチャネルＣ１、Ｃ２がＴ字形に接合する場合は、Ｔ字形の縦棒に対応するチャネルＣ１を横棒に対応するチャネルＣ２より先に配線しなければならない。チャネル幅と、左右辺から外へ出る配線の切り口の座標は、配線が完了した時点で決まる。従って、Ｔ字形の縦棒に対応するチャネルＣ１が配線完了しないと、Ｔ字形の横棒に対応するチャネルＣ２のチャネル接合部で

の端子（配線端）のＸ座標が固定されないので、チャネルＣ２を配線できない。このチャネルの配線順序制約関係を配線順序グラフで表現する。配線順序グラフは、チャネルを頂点とし、２つのチャネルがＣ１、Ｃ２Ｔ字形に接合している時Ｔ字形の縦棒のチャネルＣ１から横棒のチャネルＣ２へ向かって有向枝をつける。配線順序グラフにサイクルがある時は、サイクル内の頂点のチャネルは配線順序を決めることができず配線が不可能である。これを「チャネル配線順序のサイクル問題」[３３]という。そして、十年近く満足な方法がなかった。従来は、サイクルがある場合は配置を変更して配線領域の形状を変更し、配線順序グラフにサイクルがないチャネルの接合状態にして配線していた。第６章において配線グラフにサイクルが存在する場合の配線法を述べる。

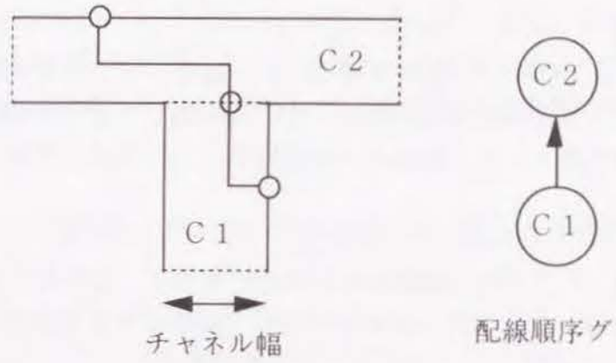


図２．１４　チャネルの配線順序

（４）スイッチボックス配線法

矩形の４辺上に端子がある配線領域を配線する。配線領域や端子位置は固定していることを前提とし、配線領域が不足すると配線不能（未配線）を発生する。主にグリーディな手法が使用されている。

（５）リバー配線法

バス信号配線を目的とした１層配線法である。バス信号のスキューを低減するために、川の流れの流線のように配線して配線長を揃える。配線を交叉させないという平面性のために、端子の信号の並び順が揃っていなければならないという強い制限がある。

第３章　最小配線長問題

３．１　はじめに

本章では、配線処理の基本である最小配線長問題を述べる。１つの信号の最小配線長の配線パターンを作成する問題は、与えられた点を最小距離でつなぐ最小スタイナ木問題になる。従来より、ＬＳＩやプリント基板は水平垂直方向に限定した２方向の直交配線である。しかし、最近のプリント基板には、線の方向を水平垂直に斜４５度を加えた水平垂直斜めの４方向配線が現われて来た。これに対応して、配線方向の数を拡張してλ方向（λは２以上の整数）にした、λ－幾何のスタイナ木が提案されている。

３．２　最小配線長問題

配線処理の目的は論理ＬＳＩのチップ面積を最小にし、論理ＬＳＩの電気特性を満たすことである。しかし、チップ面積を評価関数として直接計算しながら配線処理を実行するのは困難である。次善の策として、配線長を評価関数として配線長を最小にする配線処理がされてきた。

チップの最小配線長問題：チップ内に配線すべき信号の集合と、各信号内の接続すべき端子の位置が与えられた時、チップ内の配線長の総和を最小にする各信号の配線経路の集合を求める問題。

１信号の最小配線長問題：１つの信号の接続すべき端子の位置が与えられた時、配線長を最小にする配線経路を求める問題。

以下では、１信号の最小配線長問題について述べる。１信号の最小配線長問題によく用いられるのが最小スパンニング木と最小スタイナ木である。

３．３　スパンニング木

スパンニング木は、与えられたｎ個の点の間を（ｎ－１）個の枝で接続した木である。特に、点間を接続する枝の長さの和が最小となる木を「最小スパンニング木」と言う。

最小スパンニング木は迷路法や線分探索法に用いられる。迷路法や線分探索法は、出発点から目標点へ向かって配線経路を探索するアルゴリズムであるので。接続すべき端子位置が決まると、配線長が最小になる端子と端子の対の組合せを作って、この端子間を配線していく。即ち、最小スパンニング木を作成して、最小スパンニング木の枝に対応した端子対の間で配線経路を探索する。

最小スパンニング木を作成するアルゴリズムは、Prim[４８]とKruscal[３７]により作られた。最小スパンニング木の最大次数は、ユーグリッド幾何では６であり、直交幾何では８であり[２５]、共に有限値である。

３．４　スタイナ木

スタイナ木は、与えられた点の集合に必要に応じて新たな点（スタイナ点）を追加して、点間を接続した木である。特に、点間を接続する枝の長さの和が最小になる木を「最小スタイナ木」と言う。

スタイナ木はチャネル割当法を適用する前のグローバル配線に用いられる。セル列を積み重ねたレイアウトモデルにチャネル割当法を適用する場合には、セル列間チャネルを配線する前に信号がセル列を跨ぐ位置（フィードスルー）を予め決定しておく必要がある。このフィードスルーの位置を決める処理をグローバル配線と言う。フィードスルーの位置は配線長が最小になる位置に決める。このため、接続すべき端子位置が決まると、端子位置を点とする直交最小スタイナ木を作成して、最小スタイナ木の枝がセル列と交差する位置をフィードスルーの位置と決める。実際に詳細な配線パターンを決定するには、多数の信号の配線パターンを埋め込む必要があるので、必ずしも最小スタイナ木の形になるとは限らない。しかし、１つの信号のみを配線するとすれば配線長が最小になる。

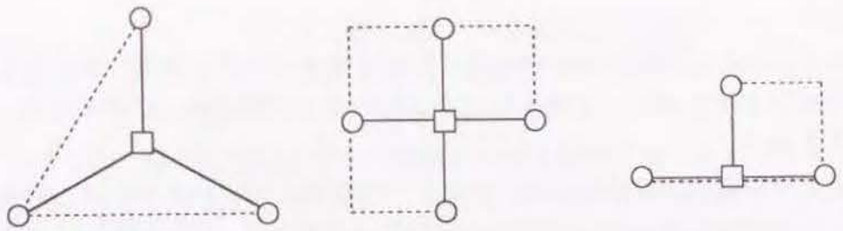
ユーグリッド幾何では、ｎ点が与えられた時、最小スタイナ木は次の性質を満たす。[６０]

（１）スタイナ点の次数は３である。
（２）与えられた点の次数は３以下である。
（３）スタイナ点の個数は（ｎ－２）以下である。
（４）スタイナ点に接続しているどの２枝も角度$2\pi/3$である。

直交幾何では、ｎ点が与えられた時、最小スタイナ木は次の性質を満たす[２１]。

（１）スタイナ点の次数は３または４である。
（２）与えられた点の次数は４以下である。
（３）スタイナ点の個数は（ｎ－２）以下である。
（４）次数４のスタイナ点に接続しているどの２枝も角度$\pi/2$である。次数３のスタイナ点に接続している２枝の角度は$\pi/2$、$\pi/2$、πである。

最小スタイナ木問題は、ユーグリッド幾何でも直交幾何でもＮＰ－completeであることが分かっている[１７][１８]。また、最小スパンニング木と最小スタイナ木の枝長和の比は、ユーグリッド幾何では高々$2/\sqrt{3}$倍である[１３]（図３．１(ａ)）。直交幾何では高々3/2倍である[２８]（（図３．１(ｂ)）。　なお、次数３では3/4倍である（図３．１(ｃ)））。最小スパンニング木は最小スタイナ木のよい近似なので、スタイナ木を作成するアルゴリズムは、最小スパンニング木を出発点として、スタイナ点を追加しながら枝の接続を変更していくヒューリスティックな方法が採用されている[２５][４２][４３]。



破線が最小スパンニング木、実線が最小スタイナ木

□ はスタイナ点

図３．１　最小スパンニング木と最小スタイナ木の最大比

３．５　λ－幾何のスタイナ木

従来、最小スタイナ木問題はユーグリッド平面と垂直水平線に限定した直交幾何の分野で研究されてきた[２５][２９][４２][４３]。特に後者はＬＳＩやプリント基板上の配線設計に重要な役割を果たしている。しかし、最近の多層プリント基板ではＸ軸、Ｙ軸方向と斜め４５度方向を用いて配線されるものがある[３８][５４]。

最近、ユーグリッド平面上と、水平垂直の直交配線の間をつなぐ概念として、「λ－幾何」の発表がされた[４][５１][５８]。λ－幾何は、線の方向として水平軸とπ/λの倍数（λは２以上の正整数）の方向（軸方向）のみに限定した幾何である（図３．２）。軸方向が水平垂直の場合は２－幾何（$\lambda=2$）、ユーグリッド平面は∞－幾何（$\lambda=\infty$）である。Ｘ軸、Ｙ軸方向と斜め４５度方向を用いて配線されるものは４－幾何（$\lambda=4$）に対応する。λ－幾何（$3\leqq\lambda<\infty$）では、最小スパンニング木と最小スタイナ木の枝長和の最大比は未だ不明である。第４章ではλ－幾何のスタイナ木の作成法について述べる。

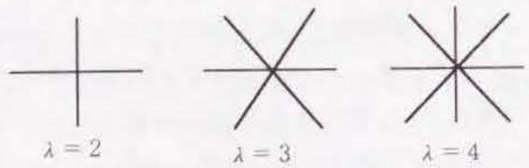


図３．２　λ－幾何の軸方向

第４章　λ－幾何のスタイナ木作成法

４．１　はじめに

本章では、、λ－幾何（$\lambda\equiv1,\ 2\ (\mathrm{mod}\ 3)$）のスタイナ木の性質を述べ、新たなスタイナ木の作成法を提案する[７３]。先ず、λ－幾何の２点間の距離の計算式を導き出す。そして、３点の最小スタイナ木のスタイナ点の位置の予想を提出し、３点が特別な位置関係にある場合にこの予想が正しいことを証明する。次に、この予想を用いて、３点及び４点のスタイナ木の幾何学的構成法を提案する。そして、与えられた任意個数の点に対する多点スタイナ木の作成法を提案する。この方法は、λ－幾何の最小スパンニング木を出発点として、３点／４点の部分木を３点／４点のスタイナ木で置き換えることを繰り返す方法である。そして、計算機を用いた実験結果について述べる。

４．２　λ－幾何のスタイナ木問題

λ－幾何は、前章で述べたように、線の方向として水平軸とπ/λの倍数（λは２以上の正整数）の方向（軸方向と呼ぶ）のみに限定した幾何である（図３．２）。先ず、λ－幾何の距離を次のように定義する。

[定義４．１]　２点p_1、p_2間を軸方向の線分l_1、l_2、・・・、l_mでつなぐ。。各線分の長さをd_iとしてΣd_iの最小値が２点p_1、p_2間の距離である。

２点p_1、p_2間の距離を実現する線分の組合せは無限にある。しかし、図４．２にしめすように、２点p_1、p_2間の距離を実現する２つの線分l_1、l_2が存在することが示された[５８]。λ－幾何の最小スタイナ木問題を次のように定義する。

［定義４．２］ユーグリッド平面上に与えられた点の集合をＰとし、必要に応じて新たな点（スタイナ点）を追加し、点間を接続した木にする。点をつなぐ枝の長さは定義４．１の距離で測るとして、枝の長さの和が最小の木を作成するのが「λ－幾何最小スタイナ木問題」である。

４．３　従来のλ－幾何のスタイナ木作成法

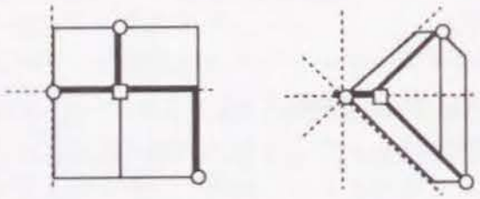
λ－幾何（$2\leqq\lambda<\infty$）に対して、スタイナ木作成法を提案したのは、M.Sarrafzadeh & C.Wong[５１]とS.Burman et al.[４]である。前者の方法は、最小スパンニング木（以下ＭＳＴと略す）の枝を節点としたbalanced binary tree Ｂpを作成して、ＢpからＭＳＴを再構成しながら枝をλ－幾何の軸方向にする方法である。後者は、ＭＳＴの枝を直接λ－幾何の軸方向にする方法である。両方法共に、枝を軸方向の２線分にする時の自由度を利用して線分の重なりを作ることによりＭＳＴより短くするのが特徴である。

従来より、２－幾何では、ＭＳＴの枝を軸方向の「Ｌ」字形の２線分にし、線分の重なり部分にスタイナ点を追加して枝長和を短縮することが基本戦略である[２５][４３]。２－幾何において、ＭＳＴの１つの点から軸方向の２線を引いてできる４個の領域のうちの隣接２領域の角度π内に枝先の２点があるとする（図４．１(a)）。各２点間の最短経路が存在する矩形を描くと、３個の矩形は相互に重なり（線も重なりとする）重な

りはすべて連結する。この重なりの中にスタイナ点と枝を設けると、各２点間は最短距離のまま３点に対する２－幾何のスタイナ木になる。この性質が、２－幾何においてＭＳＴの枝を軸方向のＬ字形の２線分にして線分の重なりを多くする方法が採用される理由である。

しかし、λ－幾何（$4 \leq \lambda < \infty$）では、この戦略は２－幾何の場合程の効果を生じない。λ－幾何では軸方向のλ線でできる隣接２領域は角度π/λと狭くなるので、角度π/λ内にＭＳＴの２枝が入ることが少なくなる。更に、２点間の最短経路が存在するのは軸方向の辺から成る平行四辺形内であるので、３個の平行四辺形の重なりを利用した枝長和の短縮はあまり期待できない（図４．１(ｂ)）。従って、λ－幾何のスタイナ木問題には枝を軸方向の２線分にするのとは別の方法が必要である。

本章で述べる方法は、枝を軸方向の２線分にする時の自由度を利用した線の重なりによる短縮よりも短くできる。



（ａ）２－幾何　　（ｂ）４－幾何

□　スタイナ点

図４．１　枝のＬ字形によるスタイナ木

４．４　λ－幾何のスタイナ木の性質

４．４．１　λ－幾何の距離計算式

本節では、λ－幾何における２点間の距離の計算式を求める。

[補題４．３]　λ－幾何における２点間の距離ｄは、

$$d = r\cos(\frac{\pi}{2\lambda} - \varepsilon)/\cos\frac{\pi}{2\lambda}$$

である。但し、ｒはユーグリッド幾何の２点間の距離、εは２点間を通る直線が直近の軸方向となす角度である。

（証明）図４．２に示すように、簡単化のために２点をａ、ｂとし、点ａの座標を原点（０、０）とし、点ｂの座標を（ｘ、ｙ）とする。λ－幾何は軸方向がπ/λの倍数のみであるので、点ｂは、点ａを中心にしてｉπ/λの軸方向と（ｉ＋１）π/λの軸方向の間にあるとする（$0 \leq i \leq 2\lambda$）。点ａ、ｂから各々ｉπ/λと（ｉ＋１）π/λの方向の２直線を引いて、２つの交点をｃ、ｄとする。

２点ａ、ｂ間の距離は、２つの線分ａｃと線分ｃｂ（または線分ａｄと線分ｄｂ）の長さの和に等しい。ここで、線分ａｃの長さをd_1、線分ｃｂの長さをd_2とする。点ｂの座標（ｘ、ｙ）は線分の長さd_1、d_2による次の連立方程式で表わされる。。

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos i\pi/\lambda & \cos(i+1)\pi/\lambda \\ \sin i\pi/\lambda & \sin(i+1)\pi/\lambda \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

この連立方程式より、d_1、d_2 を求めると、

$$d_1 = \{x\sin\frac{(i+1)\pi}{\lambda} - y\cos\frac{(i+1)\pi}{\lambda}\}/\sin\frac{\pi}{\lambda}$$

$$d_2 = \{y\cos\frac{i\pi}{\lambda} - x\cos\frac{i\pi}{\lambda}\}/\sin\frac{\pi}{\lambda}$$

である。２点ａ、ｂ間の距離ｄは、

$$d = d_1 + d_2 = \sqrt{x^2+y^2}\,\frac{\cos(\frac{\pi}{2\lambda} - \varepsilon)}{\cos\frac{\pi}{2\lambda}} \qquad \text{但し、} \varepsilon = \arctan\frac{y}{x} - \frac{i\pi}{\lambda},\ 0 \leq \varepsilon \leq \frac{\pi}{\lambda}$$

となる。εは２点ａ、ｂを通る直線が直近の軸方向となす角度である。□

λ－幾何における２点間の距離は、ユーグリッド幾何の２点間の距離$\sqrt{x^2+y^2}$と、２点を通る直線が直近の軸方向となす角度εを用いて計算することができる。$\varepsilon = 0$または$\varepsilon = \pi/\lambda$の時は、点ｂが軸方向線上にあり、距離ｄはユーグリッド幾何の距離に一致する。

以下の節では、特に断り書きがある場合を除いて、スタイナ木の枝は直線であるとし、角度は枝の直線がなす角度を測る。

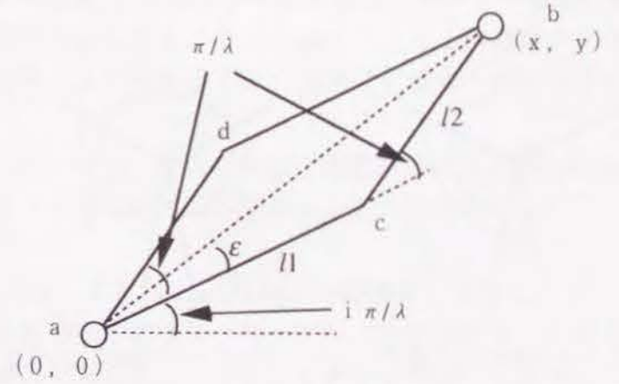


図４．２　λ－幾何の２点間の距離

４．４．２　３点スタイナ木のスタイナ点位置

３点が与えられた時の最小スタイナ木のスタイナ点の位置について述べる。λ－幾何のスタイナ点に関する次の補題がM.Sarrafzadeh & C.K.Wong [５１] により示された。

[補題４．４]　最小スタイナ木のスタイナ点の次数は３または４であり、スタイナ点の数は高々ｎ－２である。スタイナ点につながる線分は２π をできるだけ均等に分割する。

λ－幾何は軸方向がπ/λの倍数のみであるので、λが偶数の時のみ存在する次数４のスタイナ点の周りの角度はすべてπ/２である。次数３のスタイナ点の周りの角度は、$\lambda \equiv 0 \pmod 3$ の場合２π/３になる。$\lambda \equiv 1$、$2 \pmod 3$ の場合、一般に３線すべてが軸方向に一致することはできない。

以下では、$\lambda \equiv 1$、$2 \pmod 3$ の場合について述べる。２πを３等分した時の端数の角度が小さくなるように、軸方向に合わせて切り上げ、切り捨てた角度を次のように定義する。

[定義４．５]　角α，βを次のように定義する。

$\lambda \equiv 1 \pmod 3$ の時、$\alpha = \lceil 2\lambda/3 \rceil (\pi/\lambda)$，$\beta = \lfloor 2\lambda/3 \rfloor (\pi/\lambda)$。

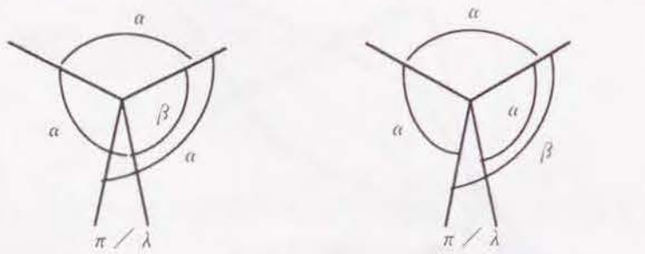
$\lambda \equiv 2 \pmod 3$ の時、$\alpha = \lfloor 2\lambda/3 \rfloor (\pi/\lambda)$，$\beta = \lceil 2\lambda/3 \rceil (\pi/\lambda)$。

言い換えると、

$\lambda \equiv 1 \pmod 3$ の時、$\alpha = (2\lambda+1)\pi/(3\lambda)$，$\beta = (2\lambda-2)\pi/(3\lambda)$。

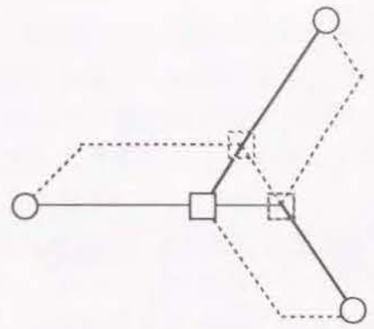
$\lambda \equiv 2 \pmod 3$ の時、$\alpha = (2\lambda-1)\pi/(3\lambda)$，$\beta = (2\lambda+2)\pi/(3\lambda)$。

この定義を用いると、スタイナ点につながる３線分（すべて軸方向）の間の角度はα，α,βと表わされる。図４．３に示すように、$\lambda \equiv 1 \pmod 3$ の時は、$3\alpha = 2\pi + (\pi/\lambda)$、$\beta = \alpha - (\pi/\lambda)$であり、$\lambda \equiv 2 \pmod 3$ の時は、$3\alpha = 2\pi - (\pi/\lambda)$、$\beta = \alpha + (\pi/\lambda)$である。例えば、λ＝４ではα＝3π/４、β＝π/２であり、λ＝５ではα＝3π/５、β＝4π/５である。



（ａ）$\lambda \equiv 1 \pmod 3$ の場合　　（ｂ）$\lambda \equiv 2 \pmod 3$ の場合

図４．３　スタイナ点の周りの角度



破線のスタイナ木もある

図４．４　λ＝３の３点の最小スタイナ木

次に、$\lambda \equiv 1$、$2 \pmod 3$ の場合に、２π をできるだけ均等に分割する点としてσ点を定義する。

[定義４．６]　次の２条件を満たす点をσ点と呼ぶ。

（１）与えられた３点とσ点を直線で結んだ時、σ点の周りの３つの角が、$\lambda \equiv 1 \pmod 3$ の時はα，$\alpha - \varepsilon_1$，$\alpha - \varepsilon_2$となる。$\lambda \equiv 2 \pmod 3$ の時はα，$\alpha + \varepsilon_1$，$\alpha + \varepsilon_2$となる。ただし、$0 \leqq \varepsilon_1, \varepsilon_2 \leqq \pi/\lambda$，$\varepsilon_1 + \varepsilon_2 = \pi/\lambda$である。

（２）角αをなす２つの直線がλ－幾何の軸方向である。

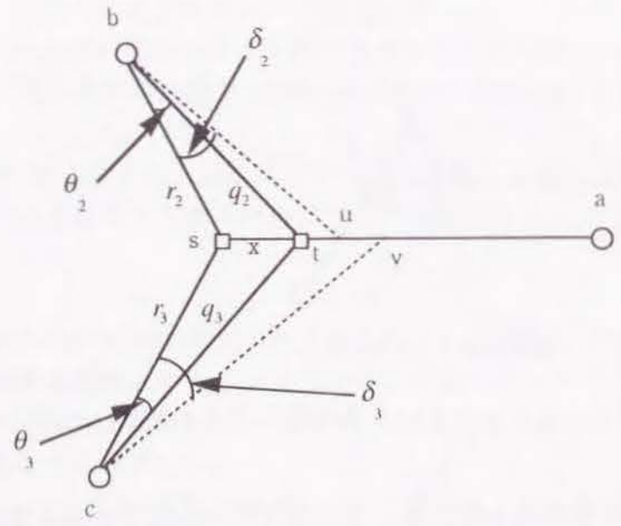
[予想４．７]　$\lambda \equiv 1$ または $\lambda \equiv 2 \pmod 3$ の場合、λ－幾何の３点の最小スタイナ木のスタイナ点はσ点である。

λ＝∞の場合、σ点の周りの３つの角度はすべてα＝２π/３になり、ユーグリッド幾何の３点の最小スタイナ木のスタイナ点とσ点は一致する。λ＝２の場合、σ点の周りの角度は２つがα＝π/２、１つがβ＝πになる。λ＝２の場合にもσ点が３点の最小スタイナ木のスタイナ点になることは次節４．５．１の最後に示す。$\lambda \equiv 0 \pmod 3$ を除いた理由は、図４．４に示すように、スタイナ点が一意に決まらない場合があるからである。

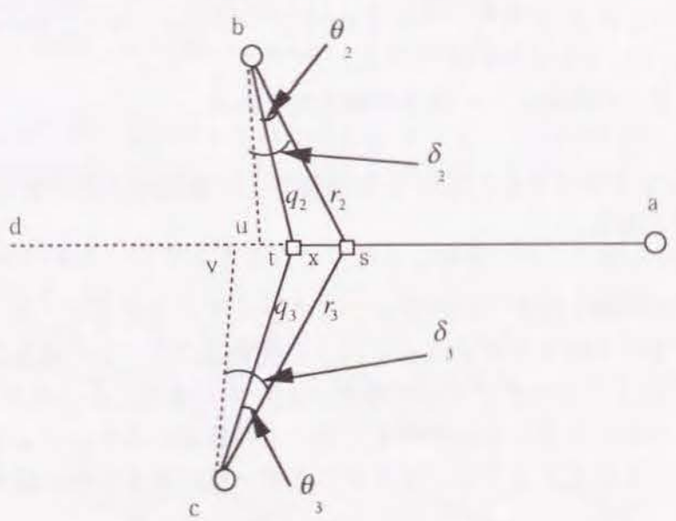
この予想４．７について、３点が特別な位置関係にある場合には、σ点が３点の最小スタイナ木のスタイナ点になることを示す。

[定理４．８]　$\lambda \equiv 1$ または $\lambda \equiv 2 \pmod 3$ の場合、３点ａ、ｂ、ｃが与えられたとする。１点ａを通る軸方向の線上に、ユーグリッド幾何のスタイナ点ｓがあり、且つ、点ａを通る軸方向線の両側に残り２点ｂ、ｃがある時は、σ点がλ－幾何の３点の最小スタイナ木のスタイナ点である。

（証明）３点ａ、ｂ、ｃが与えられた時、図４．５に示すように、１点ａを通る軸方



（a）スタイナ点ｔが点ｓと点ａの間にある場合



（b）スタイナ点ｔが点ｓと点ｄの間にある場合

図４．５　３点の最小スタイナ木のスタイナ位置

向の線に対して他の２点ｂ、ｃが線の上下にあって、ユーグリッド幾何の最小スタイナ木のスタイナ点ｓが同じ線上にあるとする。点ａは点ｂ、ｃから十分離れているものとする。点ａを通る軸方向の線が水平方向であるとして、以下この線を水平線と呼ぶ。この水平線上にλ－幾何のスタイナ点ｔを置いた場合、点ｔがσ点と一致する時に最小スタイナ木になることを示す。

３点ａ、ｂ、ｃと点ｓをつなぐ線の間の角度はすべて$2\pi/3$である。点ａと点ｓの間、点ｂと点ｓの間、点ｃと点ｓの間の各ユーグリッド距離をr_1、r_2、r_3とする。枝の長さは$r_2 \leqq r_3 < r_1$と仮定する。ユーグリッド幾何のスタイナ木の長さは（$r_1+r_2+r_3$）である。λ－幾何（$2 \leqq \lambda < \infty$）のスタイナ木の長さは（$r_1+r_2+r_3$）以上である。以下、スタイナ点ｔが点ａと点ｓの間にある場合と、スタイナ点ｔが点ｓに関して点ａの反対側にある場合に分けて述べる。

（１）スタイナ点ｔが点ａと点ｓの間にある場合。

図４．５(a)に示すように、点ｓと点ｔのユーグリッド距離をｘとする。点ｂと点ｔの間、点ｃと点ｔの間の各ユーグリッド距離をq_2、q_3とする。ユーグリッド幾何のスタイナ木の枝とλ－幾何のスタイナ木の枝との角度を$\angle sbt=\theta_2$、$\angle sct=\theta_3$とする。枝の長さは$r_2 \leqq r_3$と仮定したので$q_2 \leqq q_3$、角度は$\theta_3 \leqq \theta_2$となる。

λ－幾何の軸方向はλ/πを単位としている。点ｂを通る軸方向線の中で水平線と交差する点がｔとａの間にあって、ｔに最も近い軸方向線を考える。両軸方向線の交差する点をｕとする。λ－幾何の軸方向線は水平線と角度λ/πの倍数で交差するので、∠ｂｕａはπ/λの倍数である。∠ｂｕａ＝ｉπ/λ（ｉは正整数）と表わす。ｉは$(2\pi/3+\theta_2)/(\pi/\lambda)$より小さくない（切り上げた）整数である。そして、この軸方向線とユーグリッド幾何のスタイナ木の枝ｂｓとの角度を∠ｓｂｕ＝δ_2と表わす。δ_2＝ｉπ/λ－２π/３である。つまり、λ－幾何のスタイナ点ｔの位置を決めると、θ_2、ｉπ/λ、δ_2が決まる関係がある。

同様に、点ｃを通る軸方向線の中で、水平線と交差する点がｔとａの間にあって、ｔに最も近い軸方向線を考える。両軸方向線の交差する点をｖとする。∠ｃｖａはπ/λの倍数である。∠ｃｖａ＝ｊπ/λ（ｊは正整数）と表わす。ｊは$(2\pi/3+\theta_3)/(\pi/\lambda)$より小さくない（切り上げた）整数である。そして、この軸方向線と枝ｃｓとの角度を∠ｓｃｖ＝δ_3と表わす。δ_3＝ｊπ/λ－２π/３である。

ユーグリッド幾何ではΔsbtとΔsctにおいて次の正弦法則が成立する。

$$\frac{r_2}{\sin(\pi/3-\theta_2)}=\frac{q_2}{\sin 2\pi/3}=\frac{x}{\sin\theta_2}$$

$$\frac{r_3}{\sin(\pi/3-\theta_3)}=\frac{q_3}{\sin 2\pi/3}=\frac{x}{\sin\theta_3}$$

λ－幾何のスタイナ木の長さＬは、前節の補題４．３の２点間の距離の式に、線ｂｔ、

ｃｔと軸方向線との角度（$\delta_2-\theta_2$）と（$\delta_3-\theta_3$）を代入して、次のように表わされる。

$$L=\eta-x+q_2\frac{\cos(\pi/2\lambda-\delta_2+\theta_2)}{\cos\pi/2\lambda}+q_3\frac{\cos(\pi/2\lambda-\delta_3+\theta_3)}{\cos\pi/2\lambda}$$

$$=\eta-r_2\frac{\sin\theta_2}{\sin(\pi/3-\theta_2)}+r_2\frac{\sin 2\pi/3}{\sin(\pi/3-\theta_2)}\frac{\cos(\pi/2\lambda-\delta_2+\theta_2)}{\cos\pi/2\lambda}$$

$$+r_3\frac{\sin 2\pi/3}{\sin(\pi/3-\theta_3)}\frac{\cos(\pi/2\lambda-\delta_3+\theta_3)}{\cos\pi/2\lambda}$$

この長さＬを角度θ_2で微分する。δ_2（δ_3）はθ_2（θ_3）の関数であるが、点ｂまたは点ｃを通るλ－幾何の軸方向線と水平線と交差する点ごとの各区間内ではδ_2、δ_3は定数としてよい。先ず、正弦法則より次式が成立する。

$$\frac{r_2}{\sin^2(\pi/3-\theta_2)}=\frac{r_3}{\sin^2(\pi/3-\theta_3)}\frac{d\theta_3}{d\theta_2}$$

これを用いて、

$$\frac{dL}{d\theta_2}=r_2\frac{\sin\frac{2\pi}{3}\{2\cos(\frac{\pi}{2\lambda}+\frac{\pi}{3}-\frac{\delta_2+\delta_3}{2})\cos\frac{\delta_2-\delta_3}{2}-\cos\frac{\pi}{2\lambda}\}}{\sin^2(\frac{\pi}{3}-\theta_2)}$$

となる。点ｂまたは点ｃを通るλ－幾何の軸方向線と水平線と交差する点ごとの各区間ごとにｄＬ/ｄθ_2の値の正負を求める。

θ_2の値を大きくして行くことは、点ｔを点ｓから順に点ａの方に進めて行くことである。$\lambda\equiv 2\pmod 3$ の場合はｄＬ/ｄθ_2の値は正であり、θ_2に関して単調増加する。$\lambda\equiv 1\pmod 3$ の場合、線ｂｔが線ｂｓに最も近い軸方向線と一致するまでは、ｄＬ/ｄθ_2の値は負である。点ｔがこの交差点を越えて点ａに近づくとｄＬ/ｄθ_2の値は正になる。この交差点においてＬは極小値になる。

線ｂｔが線ｂｓに最も近い軸方向線一致する時のθ_2の値は次のように表わされる。$i\pi/\lambda\geq 2\pi/3$より、ｉは$i\geq 2\lambda/3$を満たす最小整数であるので、$i=(2\lambda+1)/\lambda$である。このｉの値をδ_2に代入して、$\theta_2=\delta_2=\pi/3\ \lambda$である。

（2）スタイナ点ｔが点ｓに関して点ａの反対側にある場合。

図4．5(b)に示すように、（1）と同様の記号を用いる。点ｓと点ｔの間のユーグリッド距離をｘとする。点ｂと点ｔの間、点ｃと点ｔの間の各ユーグリッド距離をq_2、q_3とする。角度は∠ｓｂｔ＝θ_2、∠ｓｃｔ＝θ_3とする。枝の長さは$r_2\leq r_3$と仮定したので$q_2\leq q_3$、角度は$\theta_2\leq\theta_3$となる。説明のため水平線の延長上点ｓに関して点ａの反対側の充分遠い点をｄとする。

λ－幾何の軸方向はλ/πを単位としている。点ｂを通る軸方向線の中で水平線と交差する点がｔとｄの間にあって、ｔに最も近い軸方向線を考える。両軸方向線の交差する

点をｕとする。λ－幾何の軸方向線は水平線と角度λ/πの倍数で交差するので、∠ｂｕａはπ/λの倍数である。∠ｂｕａ＝ｉπ/λ（ｉは正整数）と表わす。ｉは$(2\pi/3-\theta_2)/(\pi/\lambda)$より大きくない（切り捨てた）整数である。そして、角度を∠ｓｂｕ＝δ_2と表わす。$\delta_2=2\pi/3-i\pi/\lambda$である。つまり、λ－幾何のスタイナ点ｔの位置を決めると、θ_2、$i\pi/\lambda$、δ_2が決まる関係がある。

同様に、点ｃを通る軸方向線の中で、水平線と交差する点がｔとｄの間にあって、ｔに最も近い軸方向線を考える。両軸方向線の交差する点をｖとする。∠ｃｖａはπ/λの倍数である。∠ｃｖａ＝ｊπ/λ（ｊは正整数）と表わす。ｊは$(2\pi/3-\theta_3)/(\pi/\lambda)$より大きくない（切り捨てた）整数である。そして、角度を∠ｓｃｖ＝δ_3と表わす。$\delta_3=2\pi/3-j\pi/\lambda$である。

ユーグリッド幾何では△ｓｂｔと△ｓｃｔにおいて次の正弦法則が成立する。

$$\frac{r_2}{\sin(2\pi/3-\theta_2)}=\frac{q_2}{\sin\pi/3}=\frac{x}{\sin\theta_2}$$

$$\frac{r_3}{\sin(2\pi/3-\theta_3)}=\frac{q_3}{\sin\pi/3}=\frac{x}{\sin\theta_3}$$

λ－幾何のスタイナ木の長さＬは、次のように表わされる。

$$L=\eta+x+q_2\frac{\cos(\pi/2\lambda-\delta_2+\theta_2)}{\cos\pi/2\lambda}+q_3\frac{\cos(\pi/2\lambda-\delta_3+\theta_3)}{\cos\pi/2\lambda}$$

$$=\eta+r_2\frac{\sin\theta_2}{\sin(2\pi/3-\theta_2)}+r_2\frac{\sin\pi/3}{\sin(2\pi/3-\theta_2)}\frac{\cos(\pi/2\lambda-\delta_2+\theta_2)}{\cos\pi/2\lambda}$$

$$+r_3\frac{\sin\pi/3}{\sin(2\pi/3-\theta_3)}\frac{\cos(\pi/2\lambda-\delta_3+\theta_3)}{\cos\pi/2\lambda}$$

この長さＬを角度θ_2で微分する。δ_2（δ_3）はθ_2（θ_3）の関数であるが、点ｂまたは点ｃを通るλ－幾何の軸方向線と水平線と交差する点ごとの各区間内ではδ_2、δ_3は定数としてよい。先ず、正弦法則より次式が成立する。

$$\frac{r_2}{\sin^2(2\pi/3-\theta_2)}=\frac{r_3}{\sin^2(2\pi/3-\theta_3)}\frac{d\theta_3}{d\theta_2}$$

これを用いて、

$$\frac{dL}{d\theta_2}=r_2\frac{\sin\frac{\pi}{3}\{2\cos(\frac{\pi}{2\lambda}+\frac{2\pi}{3}-\frac{\delta_2+\delta_3}{2})\cos\frac{\delta_2-\delta_3}{2}+\cos\frac{\pi}{2\lambda}\}}{\sin^2(\frac{2\pi}{3}-\theta_2)}$$

となる。点ｂまたは点ｃを通るλ－幾何の軸方向線と水平線と交差する点ごとの各区間ごとにｄＬ/ｄθ_2の値の正負を求める。

θ_2の値を大きくして行くことは、点ｔを点ｓから順に点ｄの方に進めて行くことである。$\lambda\equiv 1\pmod 3$ の場合はｄＬ/ｄθ_2の値は正であり、θ_2に関して単調増加する。

$\lambda \equiv 2 \pmod 3$ の場合、線ｂｔがユーグリッド幾何のスタイナ木の枝に最も近い軸方向線と一致するまでは、ｄＬ/ｄθ_2の値は負である。この交差点を越えて点ｄに近づくとｄＬ/ｄθ_2の値は正になる。この交差点においてＬは極小値になる。

線ｂｔがユーグリッド幾何のスタイナ木の枝に最も近い軸方向線一致する時のθ_2の値は次のように表わされる。ｉ$\pi/\lambda \leqq 2\pi/3$より、ｉはｉ$\leqq 2\lambda/3$を満たす最大整数であるので、ｉ$=(2\lambda-1)/\lambda$である。このｉの値をδ_2に代入して、$\theta_2=\delta_2=2\pi/3\lambda$である。

以上（１）と（２）より、$\lambda \equiv 1 \pmod 3$の場合は、λ一幾何のスタイナ木の枝ｂｔが、点ｂを通り線ｂｓに最も近い∠ｂｓａ側の軸方向線と一致した時に、Ｌは最小値になる。この時のλ一幾何のスタイナ点の周りの３つの角度は次のようになる。

∠ｂｔａ$=\alpha=(2\lambda+1)\pi/(3\lambda)$。

$\beta=(2\lambda-2)\pi/3 \leqq$∠ｃｔａ$\leqq\alpha=\lambda(2\lambda+1)\pi/(3\lambda)$。

$\beta=(2\lambda-2)\pi/3 \leqq$∠ｂｔｃ$\leqq\alpha=\lambda(2\lambda+1)\pi/(3\lambda)$。

即ち、３つの角は、$\alpha, \alpha-\varepsilon_1, \alpha-\varepsilon_2$である。ただし、$0 \leqq \varepsilon_1, \varepsilon_2 \leqq \pi/\lambda$，$\varepsilon_1+\varepsilon_2=\pi/\lambda$である。そして、角$\alpha$をなす２つの直線はλ一幾何の軸方向である。即ち、Ｌが最小となるスタイナ点位置はσ点である。

$\lambda \equiv 2 \pmod 3$の場合は、λ一幾何のスタイナ木の枝ｂｔが、点ｂを通り線ｂｓに最も近い∠ｂｓｄ側の軸方向線一致した時に、Ｌは最小値になる。この時のλ一幾何のスタイナ点の周りの３つの角度は次のようになる。

∠ｂｔａ$=\alpha=(2\lambda-1)\pi/(3\lambda)$。

$\alpha=(2\lambda-1)\pi/3 \leqq$∠ｃｔａ$\leqq\beta=\lambda(2\lambda+2)\pi/(3\lambda)$。

$\alpha=(2\lambda-1)\pi/3 \leqq$∠ｂｔｃ$\leqq\beta=\lambda(2\lambda+2)\pi/(3\lambda)$。

即ち、３つの角は、$\alpha, \alpha+\varepsilon_1, \alpha+\varepsilon_2$となる。ただし、$0 \leqq \varepsilon_1, \varepsilon_2 \leqq \pi/\lambda$，$\varepsilon_1+\varepsilon_2=\pi/\lambda$である。そして、角$\alpha$をなす２つの直線はλ一幾何の軸方向である。即ち、Ｌが最小となるスタイナ点位置はσ点である。

故に、スタイナ点ｔがσ点と一致する時に最小スタイナ木になる。□

４．５　λ一幾何のスタイナ木作成法

４．５．１　３点スタイナ木の作成法

λ一幾何の３点スタイナ木の作成法のアイデアは、前節４．４．２で述べたσ点にスタイナ点を置くことである。３点スタイナ木の幾何学的作成手段として、２点で決まる図形を次のように定義する（図４．６）。

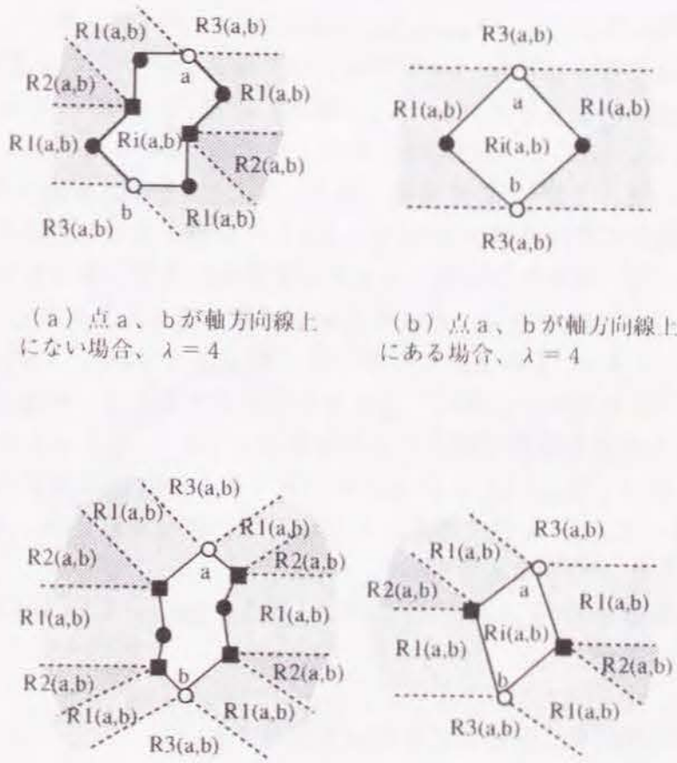
[定義４．９]　２点を通る軸方向の２線が角αで交差する点をα点と呼ぶ。また、$4 \leqq \lambda < \infty$に対して、２点を通る軸方向の２線が角βで交差する点をβ点と呼ぶ。

$\lambda=2$の場合は、$\beta=\pi$になるので、β点はない。

[定義４．１０]　与えられた２点ａ、ｂとα点とβ点を周回しながら結んでできる多角形を「２点ａ、ｂで決まる周囲多角形」と呼び、Ｅｎ(ａ，ｂ)と表わす。Ｅｎ(ａ，ｂ)の辺は全て軸方向の線分である。

Ｅｎ(ａ，ｂ)とその内部領域を合わせて「２点ａ、ｂで決まる内部領域」と呼び、Ｒｉ(ａ，ｂ)と表わす。

Ｅｎ(ａ，ｂ)の外部領域を次の３つの領域に分割する。タイプ１領域Ｒ１(ａ，ｂ)は次のように定義する。Ｅｎ(ａ，ｂ)上の任意の点をｙとした時、Ｅｎ(ａ，ｂ)の外部領域内の点ｘで、角ａｙｘまたは角ｂｙｘがαとなる点ｘの集合をＲ１(ａ，ｂ)とする。但し、線ｘｙは軸方向でＥｎ(ａ，ｂ)と交差しないものとする。Ｒ１(ａ，ｂ)はＥｎ



（ａ）点ａ、ｂが軸方向線上にない場合、$\lambda=4$

（ｂ）点ａ、ｂが軸方向線上にある場合、$\lambda=4$

（ｃ）点ａ、ｂが軸方向線上にない場合、$\lambda=5$

（ｄ）点ａ、ｂが軸方向線上にある場合、$\lambda=5$

○ 与えられた点　■ α点　● β点

図４．６　２点により決まる領域

(a，b)と2つの平行線で三方を囲まれた開領域である。

タイプ2領域R2(a，b)は、タイプ1領域にはさまれてα点から角度π/λで放射状に広がる領域である。タイプ1領域とタイプ2領域の境界線はタイプ2領域に含める。

タイプ3領域R3(a，b)は、タイプ1領域にはさまれて点a、bから放射状に広がる残った2つの領域である。タイプ3領域とタイプ1領域の境界線はタイプ3領域に含める。

以下では、En(a，b)、Ri(a，b)、R1(a，b)、R2(a，b)、R3(a，b)の2点a、bを省略して、En、Ri、R1、R2、R3と表すこともある。

σ点の定義より、σ点はα点であるので、α点の中からスタイナ点の位置を、次の2つの場合に探せばよい。

(1) 第3点cがR2(a，b)内にある場合（図4．7(b)）；点cを含むR2(a，b)が放射状に広がる根本のEn(a，b)上のα点を点sと表わす。2つの線asとbsがなす角はαであり、2つの線は共に軸方向である。線as（bs）と点a（b）に近いR2(a，b)の境界の角は、$\lambda \equiv 1 \pmod 3$の時$\beta = \alpha - (\pi/\lambda)$であり、$\lambda \equiv 2 \pmod 3$の時αである。α点でのR2(a，b)の角はπ/λである。点cと点sを直線で結ぶと、$\angle csa = \alpha \pm \varepsilon_1$、$\angle asb = \alpha \pm \varepsilon_2$である。ただし、$0 \leqq \varepsilon_1$、$\varepsilon_2 \leqq \pi/\lambda$、$\varepsilon_1 + \varepsilon_2 = \pi/\lambda$。従って、En(a，b)上のα点はσ点である。

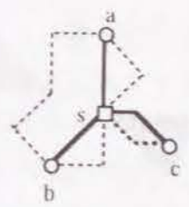
(2) 第3点cがR1(a，b)内にある場合（図4．7(a)）；点cを通ってR1(a，b)とR2(a，b)の境界に平行な線とEn(a，b)との交点を点sと表わす。R1(a，b)の定義により、角csbまたは角csaがαである。先ず、角csbがαならば2つの線bsとcsは軸方向である。点aと点sを直線で結んだとすると、$\angle csa = \alpha \pm \varepsilon_1$、$\angle asb = \alpha \pm \varepsilon_2$である。したがって、点aはR2(b，c)内にあり、点sはEn(b，c)上のα点である。次に、$\angle csa = \alpha$とすると2つの線csとasは軸方向である。点bと点sを直線で結ぶと$\angle csb = \alpha \pm \varepsilon_1$、$\angle asb = \alpha \pm \varepsilon_2$である。ただし、$0 \leqq \varepsilon_1$、$\varepsilon_2 \leqq \pi/\lambda$、$\varepsilon_1 + \varepsilon_2 = \pi/\lambda$である。点bはR2(a，c)内にあり、点sはEn(b，C)上のα点である。したがって、点sはEn(b，C)またはEn(a，C)のα点であり、同時にσ点である。

α点は3つの周囲多角形En(a，b)、En(b，C)とEn(a，C)上にのみあるので、これら以外にα点はない。

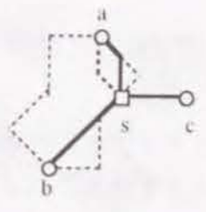
3点のスタイナ木の作成法をC言語様式を用いて記述する。

3点スタイナ木作成作成法
　(入力は3点a，b，cの座標) ｜
　Pの任意の2点a、bに対して、R1(a，b)とR2(a，b)とR3(a，b)を計算する；
　switch(第3点cが存在する位置) ｜
　　case　点cがR2(a，b)にある：

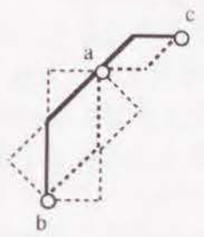
　スタイナ点を点cを含むR2(a，b)のα点に置く；
　スタイナ点と3点a、b、cと直線で結ぶ（図4．7(a)）；
　case　点cがR1(a，b)にある：｜
　点cを通ってR1(a，b)とR2(a，b)の境界に平行な線がEn(a，b)と交差する点にスタイナ点を置く；
　スタイナ点と3点a、b、cと直線で結ぶ。（図4．7(b)）；｜
　case　点cがR3(a，b)にある：
　λ－幾何の3点a、b、cの最小スパンニング木にする（図4．7(c)）；
　case　点cがRi(a，b)にある：
　λ－幾何の3点a、b、cの最小スパンニング木にする（図4．7(d)）；
｜
軸方向でない線は角π－(π/λ)をなす軸方向の2線分で置き換える。
｜



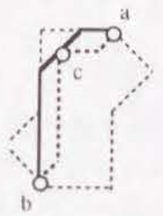
（a）点cがR2(a、b)内



（b）点cがR1(a、b)内



（c）点cがR3(a、b)内

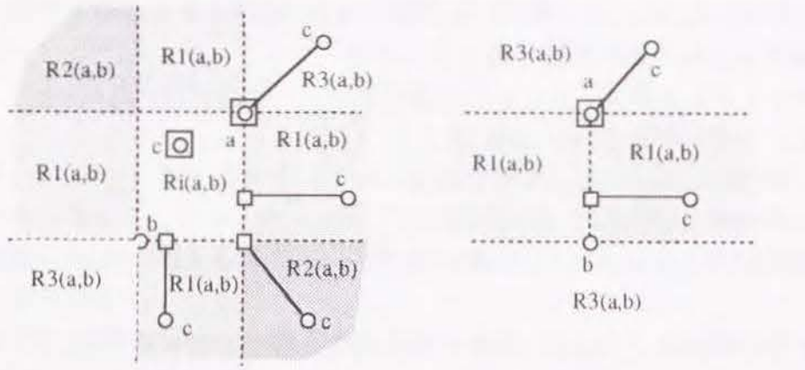


（d）点cがRi(a、b)内

図4．7　3点のスタイナ木（λ=4）

前節4．4．2で予告した、λ＝2の場合にもσ点が3点の最小スタイナ木のスタイナ点になることを示す。λ＝2の場合、3点の最小スタイナ木のスタイナ点の座標は、x座標が3点のx座標のメディアン、y座標も3点のy座標のメディアンであることが知られている[21]。本方法で作成した2－幾何の3点スタイナ木も最小スタイナ木に一致することを示す。λ＝2の場合、3点a、b、cが与えられた時、En、Ri、R1、R2、R3は図4．8の形になる。図中に第3点cの位置とその時のスタイナ点を

実線でむすんで表示した。各々スタイナ点はα点の条件を満たし、３点ａ、ｂ、ｃのｘ座標、ｙ座標のメディアンになっている。



（ａ）２点２、ｂが軸方向上でない場合　　（ｂ）２点２、ｂが軸方向上の場合

図４．８　λ＝２の場合のスタイナ点の位置

４．５．２　４点スタイナ木の作成法

本節では、４点のスタイナ木の作成法について述べる。４点の場合、最小スパンニング木と異なる最小スタイナ木のトポロジーはＹの縦棒を合わせた形と十字形の２種である。２－幾何の４点最小スタイナ木の作成法はHannan [２１]が発表した。λ－幾何の最小スタイナ木のスタイナ点の周りの角２πは均等に分割されるので、次数３のスタイナ点を高々２個置いて、各スタイナ点の周りの角を、$\lambda \equiv 1 \pmod 3$の時 α, $\alpha-\varepsilon_1$, $\alpha-\varepsilon_2$ に、$\lambda \equiv 2 \pmod 3$の時 α, $\alpha+\varepsilon_1$, $\alpha+\varepsilon_2$ にする方法と、次数４のスタイナ点（λが偶数の時のみ存在し得る）を１個置いてスタイナ点の周りの４つの角度をπ/２にする方法を組合わせた方法を述べる。ただし、本方法で作成される４点スタイナ木は、最小スタイナ木であるかどうかはまだ分からない。

この４点スタイナ木を作るために、定義４．９で述べたタイプ１領域Ｒ１とタイプ２領域Ｒ２の各々の重なりと、次に定義するπ/２点を用いる。

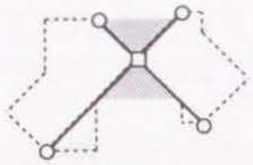
[定義４．１１]　２点を通る軸方向の２線が角π/２で交差する点をπ/２点と呼ぶ。

π/２点は、λ＝２の時α点と、λ＝４の時β点と一致する。

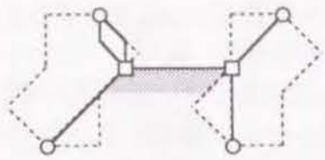
４点でできる４角形より対向２辺を取り出し、辺の両端２点ごとに、２点で決まるπ/２点、Ｒ１、Ｒ２を作る。（以下、「辺の両端の２点で決まるπ/２点、Ｒ１、Ｒ２」と言う代わりに「辺のπ/２点、Ｒ１、Ｒ２」等と言う。）そして、２つの辺のπ/２点、Ｒ１、Ｒ２の重なりの組合せにより、４点のスタイナ木の形を決める。

４点のスタイナ木の作成法をＣ言語様式を用いて記述する。

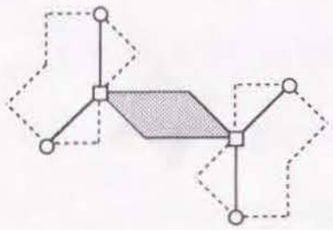

```
４点スタイナ木作成法
（入力は４点の座標）｛
ｆｏｒ（対向辺の組合せ２組について）｛
 ｉｆ（２辺のＥｎが重ならない）｛
  ２辺のπ/２点と各領域Ｒ１、Ｒ２、Ｒ３を求める；
  ｉｆ（２辺のπ/２点が重なる）｛
   π/２点にスタイナ点を置いて、４点と枝をつなぐ（図４．９（ａ））；｝
  ｅｌｓｅ｛
   ｓｗｉｔｃｈ（領域の組合せ）｛
    ｃａｓｅ ２つのＥｎからＲ１が反対方向に延びて重なり合う：
     ｉｆ（重なった領域内にβ点を含む）｛
      β点よりＲ１とＲ２の境界線に平行な線を引き、Ｅｎとの交点とβ点
      にスタイナ点を置く；
      各スタイナ点から辺端の２点へ枝をつなぐ；｝
     ｅｌｓｅ｛
      Ｒ１の重なった領域のＲ１とＲ２の境界線に線を引き、Ｅｎとの交点
      とα点にスタイナ点を置く；
      各スタイナ点から辺端の２点へ枝をつなぐ（図４．９(ｂ)）；｝
    ｃａｓｅ ２つのＲ２が互いにα点を含む：
     両方のα点にスタイナ点を置いて枝をつなぐ；
     各スタイナ点から辺端の２点へ枝をつなぐ（図４．９(ｃ)）；
    ｃａｓｅ Ｒ１もＲ２も重ならず、且つ、Ｒ２の中に辺端の点を含む：
     α点にスタイナ点を置いてＲ３の点と枝をつなぐ；
     スタイナ点から辺端の２点へ枝をつなぐ（図４．９(ｄ)）；
    ｃａｓｅ Ｒ１が重ならず、且つ、Ｒ１の中に辺端の点を含む：
     Ｒ３の点よりＲ１と平行線を引き、Ｅｎとの交点にスタイナ点を置く；
     スタイナ点から辺端の２点へ枝をつなぐ（図４．９(ｅ)）；
    ｃａｓｅ ２つのＲ３が互いに辺端の点を含む：
     λ－幾何の４点のスパンニング木とする（(図４．９（ｆ））；
   ｝
  ｝
 ｝
 対向辺の組合せ２組の結果より長さの小さい方を採用する；
 軸方向でない線は角π－(π/λ)をなす軸方向の２線分で置き換える。；
｝
```



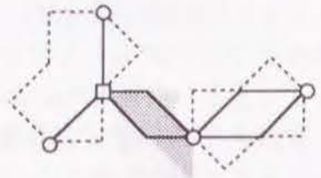
（a）π／２点が重なる場合



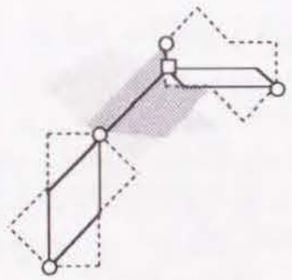
（b）Ｒ１が重なり合う場合



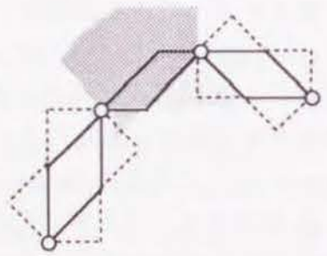
（c）Ｒ２が互いにα点を含む場合



（d）Ｒ１もＲ２も重ならず、
Ｒ２の中に辺端の点を含む場合



（e）Ｒ１が重ならず、
Ｒ１の中に辺端の点を含む場合



（f）Ｒ３が互いに辺端の点を含む場合

図４．９　４点スタイナ木（λ＝４）

４．５．３　多点スタイナ木の作成法

本節では、与えられた点集合Ｐのスタイナ木を作成する新しい方法を提案する。本方法は、λ－幾何の最小スパンニング木を出発点として、３点／４点の部分木を３点／４点のスタイナ木で置き換えることを繰り返す。λ－幾何の最小スパンニング木は、点の数をｎとして時間複雑度Ｏ(ｎlog ｎ)で得られる[５８]。

３点ａ、ｂ、ｃに対して、点ｃがＲ１(ａ，ｂ)またはＲ２(ａ，ｂ)内ならば、角ａｂｃまたは角ｂａｃはγより小である。但し、$\lambda \equiv 1 \pmod 3$の時$\gamma = \alpha$、$\lambda \equiv 2 \pmod 3$

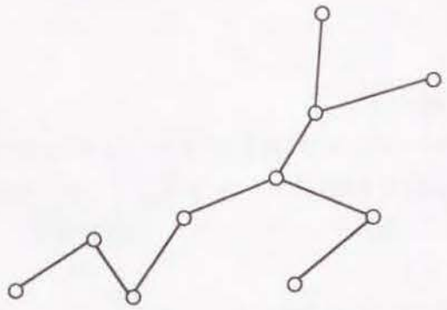
の時$\gamma = \beta$である。多点スタイナ木作成法は、３点を結ぶ２つの線のなす角がγより小さいＶ字形を含む最大５点の部分木を探し、前節４．５．１で述べた３点／４点スタイナ木作成法を適用する。３点／４点スタイナ木が３点／４点の部分木より小さければ３点／４点スタイナ木に置き換える。従って、結果のスタイナ木は最初の最小スパンニング木より長くなることはない。

多点スタイナ木の作成法をＣ言語様式を用いて記述する。

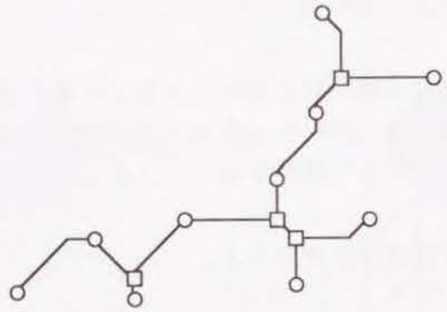
```
多点スタイナ木の作成法
（引数は点集合Ｐの座標）｜
点集合Ｐのλ－幾何の最小スパンニング木Ｔを作る。Ｔの各枝は直線にする。
Ｔの各点につながる枝を時計回りの順にソートする。
ｆｏｒ（Ｔの各点）｜
  ｆｏｒ（点につながる枝）｜
    ｉｆ（連続する２枝の角がγより小）｜
      ２枝の枝先２点より各々更に１枝先の２点を調べ、連結した最大５点を求める。
      ｓｗｉｔｃｈ（取り出した点の数）｜
        ｃａｓｅ ５点：｜
            ５点のうちの連続４点に対して４点スタイナ木作成法を適用する；
            もう一つの連続４点に対して４点スタイナ木作成法を適用する；
            枝長の短縮が大きい方を採用する；｜
          ｃａｓｅ ４点：
            ４点スタイナ木作成法を適用する；
          ｃａｓｅ ３点：
            ３点スタイナ木作成法を適用する；
      ｜
      連続する３枝より４点を求める；
      ４点スタイナ木作成法を適用する；
      ｉｆ（いずれかの４点／３点スタイナ木が４点／３点の部分木よりも短い）
        部分木を最も小さい４点／３点スタイナ木に置き換え、Ｔを変更する；
        変更した点の枝を時計回りの順にソートする；
      ｜
    ｜
  ｜
  ｆｏｒ（軸方向でない線）｜
    軸方向でない線を角π－(π/λ)をなす軸方向の２線分で置き換える；
  ｜
｜
```

ただし、上記の多点スタイナ木作成法内の３点スタイナ木作成法と４点スタイナ木作成法では、軸方向でない線は直線のままにしておく。

図４．１０に４－幾何のスタイナ木の作成例を示す。最初に最小スパンニング木（ａ）を作成し、上記の多点スタイナ木作成法を適用して、スタイナ木（ｂ）ができる。



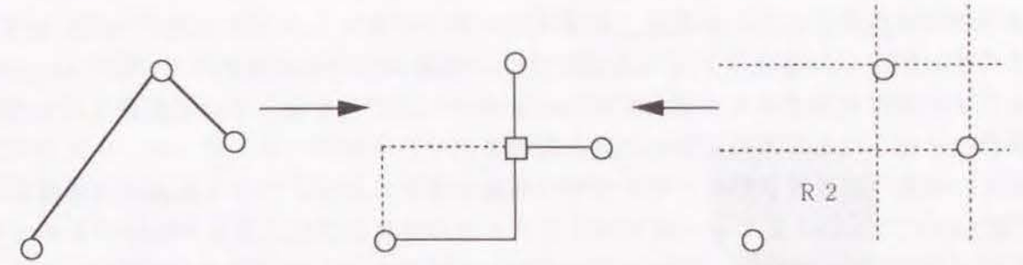
（ａ）最小スパンニング木



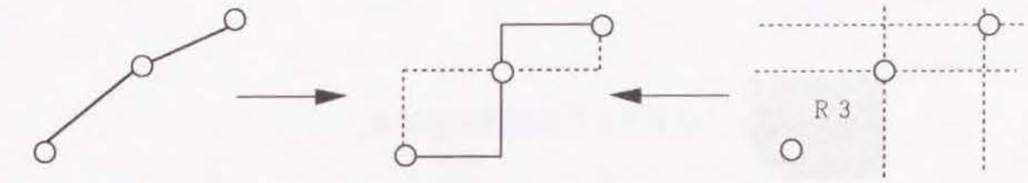
（ｂ）スタイナ木

図４．１０　スタイナ木の例（$\lambda = 4$）

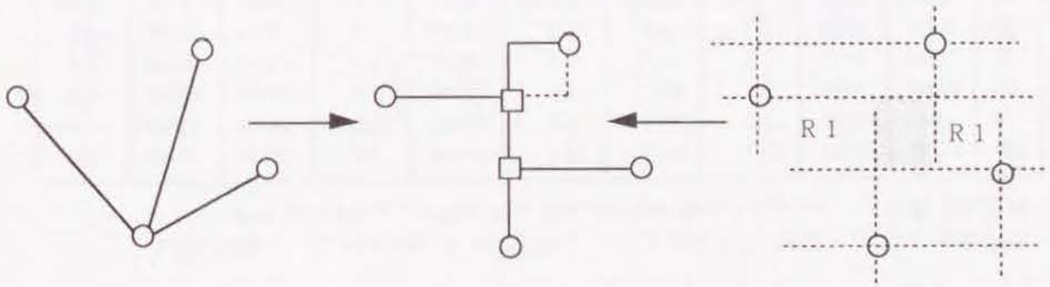
本方法はLee & Sechenの方法[４３]のλ－幾何への拡張になっていることを図４．１１に示す。図４．１１(ａ)(ｄ)(ｅ)は$\lambda = 2$に対するLee & Sechenの方法である。最小スパンニング木の１点を通る水平線または垂直線の片側に枝で結合した２点／３点があれば、３点／４点スタイナ木(ｂ)(ｅ)(ｈ)を作る。本方法では(ｃ)(ｆ)(ｉ)のように領域Ｒ１、Ｒ２、Ｒ３の中にあるか、またはこれらの組合せでスタイナ木を作る。



（ａ）水平線より上に２点　（ｂ）スタイナ木　（ｃ）第３点がＲ２内



（ｄ）水平線の上下にあり、垂直線の左右にある。　（ｅ）スタイナ木　（ｆ）第３点がＲ３内



（ｇ）水平線より上に３点　（ｈ）スタイナ木　（ｉ）Ｒ１が重なる。

図４．１１　Lee & Sechen [40] と本方法の対応

４．６　評価結果

４－幾何のスタイナ木作成法をSun Spark １０model ４０上にＣ言語で実現し、次の２項目の実験評価をした結果を表１に示す。評価は、乱数を用いて点座標を発生させ、点数ごとに１００回の平均値を求めた。

（１）提案手法によるスタイナ木の最小スパンニング木からの短縮率：

提案手法にて作成した４－幾何のスタイナ木の枝長和と４－幾何の最小スパンニング

木の枝長和を比較した。この結果、最小スパンニング木から３.２％短縮できた。変更を３点木に限定した場合は３.０％短縮できた。両者の短縮率の差が小さいことは、４点木によらねば短縮できない個所が意外に少なかったことを示している。表１の処理時間は最小スパンニング木作成後の処理の時間を示す。

（２）従来の垂直水平配線に斜め方向の配線を追加したことによる配線長の短縮率：

文献[４３]の手法による２－幾何のスタイナ木と提案手法による４－幾何のスタイナ木の枝長和を比較した。４－幾何は２－幾何より枝長和が１０.３％短縮された。これにより、斜め方向の配線を追加することは配線長を１０％程短縮する大きな効果があることが確認できた。また、S.Burman et al.[４]の短縮率は平均８．７％であるので、本方法の方が効果が大きい。

表１．提案手法の計算機実験結果

点数	４－幾何							２－幾何		
	ＭＳＴ	３点木と４点木による短縮				３点木のみ		ＭＳＴ	ＳＭＴ	短縮率
		ＳＭＴ	短縮率	ＳＭＴ比	処理時間	ＳＭＴ	短縮率			
10	4384	4243	3.3%	89.5%	0.7 ms	4251	3.1%	5163	4737	8.3%
20	6421	6209	3.3	90.0	1.6	6225	3.1	7591	6897	9.2
30	7810	7552	3.3	89.7	2.3	7563	3.2	9229	8415	8.8
50	10060	9748	3.1	89.7	4.9	9770	2.9	11934	10865	9.0
70	11943	11565	3.2	89.9	6.9	11588	3.0	14164	12863	9.2
100	14218	13765	3.2	89.7	12.1	13790	3.0	16870	15348	9.0

ＭＳＴは最小スパンニング木の枝長和　　ＳＭＴは最適スタイナ木の枝長和
短縮率は（ＭＳＴ－ＳＭＴ）／ＭＳＴ　　ＳＭＴ比は４-幾何ＳＭＴ／２-幾何ＳＭＴ

４．７　本章のまとめ

（１）λ－幾何における２点間の距離の計算式を導いた。λ－幾何における２点間の距離は、ユーグリッド幾何の２点間の距離と、２点を通る直線が直近の軸方向となす角度を用いて計算することができる。

（２）$\lambda \equiv 1$、$2 \pmod 3$ の場合に、2π をできるだけ均等に分割する点としてσ点を定義して、$\lambda \equiv 1$ または $\lambda \equiv 2 \pmod 3$ の場合、λ－幾何の３点の最小スタイナ木のスタイナ点はσ点であることを予想した。$\lambda = \infty$ のと $\lambda = 2$ の場合、既知の３点の最小スタイナ木のスタイナ点とσ点は一致することを示し、３点が特別な位置関係にある場合、σ点が３点の最小スタイナ木のスタイナ点になることを証明した。

（３）２点で決まるλ－幾何の図形を用いた幾何学的手段を開発して、３点スタイナ木及び４点スタイナの作成法を開発した。λ－幾何の最小スパンニング木を出発点として、３点／４点の部分木を３点／４点のスタイナ木で置き換えることを繰り返す、与えられた点集合Ｐの多点スタイナ木の作成方法を開発した。本方法は直交幾何の Lee & Sechenの方法[４３]のλ－幾何への拡張になっている。

（４）４－幾何のスタイナ木作成法をSun Sparc １０model ４０上にＣ言語で実現し、実験評価をした。評価は、乱数を用いて点座標を発生させ、点数ごとに１００回の平均値を求めた。提案手法にて作成した４－幾何のスタイナ木の枝長和と４－幾何の最小スパンニング木の枝長和を比較して、最小スパンニング木から３.２％短縮できた。変更を３点木に限定した場合は３.０％短縮できた。また、従来の垂直水平配線に斜め方向の配線を追加することは配線長を１０％程短縮する大きな効果があることを示した。また、S.Burman et al.[４]の短縮率は平均８．７％であるので、本方法の方が効果が大きい。

第５章　標準セルとマクロセル混在ブロック内配置配線

５．１　はじめに

本章では、標準セルとマクロセルというサイズの差異が大きいセルを混在して配置配線する方法を述べる。論理ＬＳＩは、チップ、ブロック、セルの３階層で階層レイアウト設計する。本章で対象とするのは、セルを数十～数百個配置配線して作られるブロックである。従来の自動レイアウトの対象となる標準セルはサイズ、形、共に大きな差異はない。しかし、ＲＯＭ、ＲＡＭ、ＰＬＡ、などのマクロセルは、標準セルよりサイズが大きく、セルの側面にも端子がある。このため、両者を混在して、セルのサイズの差異による無駄領域を削減するレイアウトモデルを提案し、このレイアウトモデルに基づくセルの配置方法と３辺以上に端子があるマクロセルの配線方法について述べる[７０]。

５．２　標準セルとマクロセル混在配置配線問題

従来の自動配置配線手法は標準セルのみを対象にしていた。一方で、更にチップ面積を小さくするために実装密度を上げたいという要求があり、ＲＯＭ、ＲＡＭ、ＰＬＡ、ＡＬＵなどの人手で作成したマクロセルを用いるとチップの実装密度が大きくなることが報告されている[１４][３２]。また、論理修正のマスク数を削減するためにＰＬＡを導入したいという要求があった。

標準セルとマクロセルを混在させるには、次の２つの問題を解決する必要がある。

（１）標準セルとマクロセルはサイズの差異が数～数十倍あるので、標準セルの列の中に入れた配置では無駄な領域を生じる。この無駄な領域を削減する配置ができること。

（２）標準セルは矩形の上下辺のみに端子がある。しかし、マクロセルは矩形の側面にも端子がある。この矩形の３辺以上にある端子と配線できること。

５．３　標準セルとマクロセル混在レイアウトモデル

先ず、標準セルとマクロセルの形状を定義する。セルモデルを図５．１に示す。「標準セル」は、形状は矩形で、サイズ（幅と高さ）は可変であるが高さはほぼ同じであり、端子がセルの上辺と下辺にあるセルである。「マクロセル」は、形状は矩形で、サイズ（幅と高さ）は可変であり、端子がセルの上辺と下辺のみならず左右の側辺にもあるセルである。両者のセルの主な違いは、サイズの大きさと端子のある辺の数である。

次に、ブロックの形状を定義する。従来のブロックは標準セルのみをアレーに配置配線した[３６][６７][６８]。これを拡張して、図５．２に標準セルとマクロセルを混在配置した拡張ブロックモデルを示す。「サブブロック」は、標準セル列とマクロセルを積み重ねた配置、または、標準セル列のみを積み重ねた配置である。「拡張ブロック」は複数のサブブロックを１次元に並べたブロックである。拡張ブロックモデルは２種類のサブブロックから成る。標準セル列のみを積み重ねたサブブロックは１つである。また、比較的小さなマクロセルは標準セル列の中に配置する。一方のサブブロックには核となるセルが唯一つ入っている。他方のサブブロックには核になるセルが入っていない。但

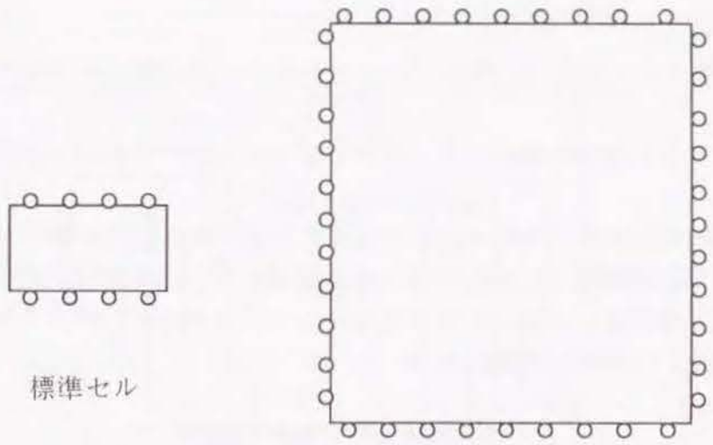


図５．１　セルモデル

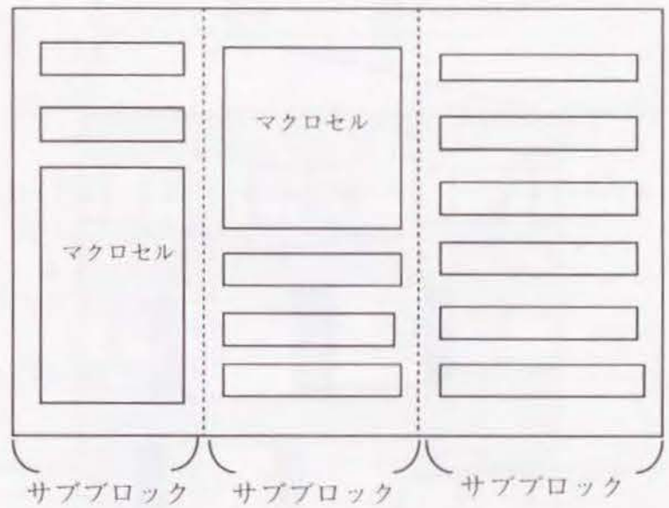


図５．２　拡張ブロックモデル

し、核になるセルとはサブブロックのシード（種）にするために指定するセルである。通常、核になるセルはマクロセルである。各サブブロックのＸサイズは核になるセルの幅で決まり、Ｙサイズはセル列数で決まる。

最後に、に３種類のチャネルを定義する。チャネルのモデルを図５．３に示す。

（ａ）「セル隙間チャネル」は、サブブロック内のマクロセルと隣接セルの間の垂直

方向のチャネルである。

（b）「セル列間チャネル」は、サブブロック内のセル列の間の水平方向のチャネルである。

（c）「サブブロック間チャネル」は、サブブロックとサブブロックの間の垂直方向のチャネルである。

サブブロック内のセル隙間チャネルとセル列間チャネルは部分的に領域が重なる。セル隙間チャネルでは側面端子からの水平線分のみ配線して、水平配線の端に「引き出し端子」と呼ぶ仮端子をおく。ブロックサイズを小さくするために、チャネルの重なり領域の線分はセル列間チャネルにて配線する。

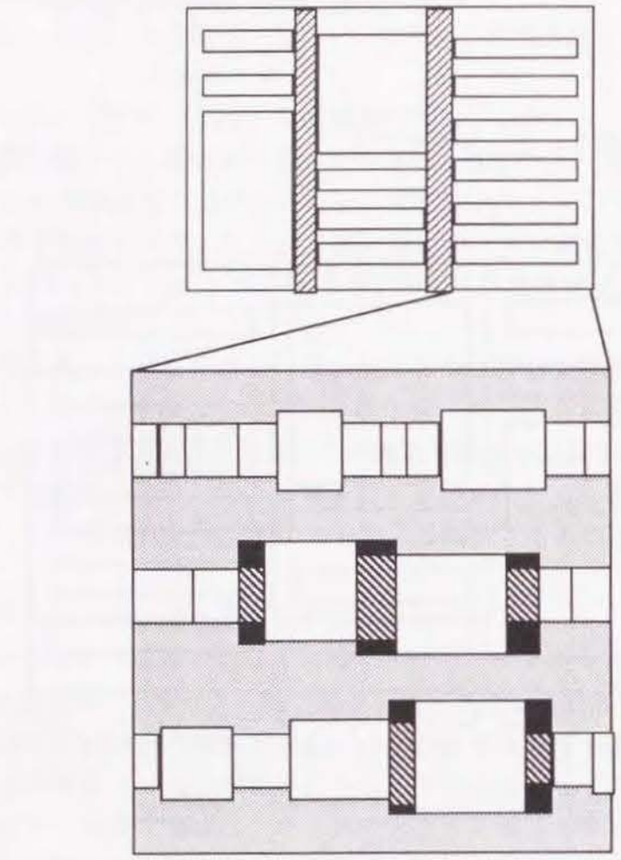
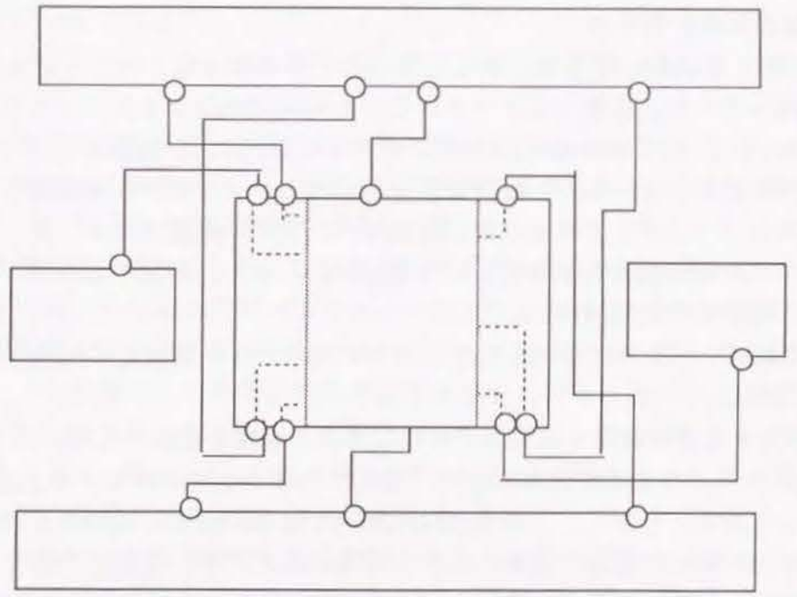
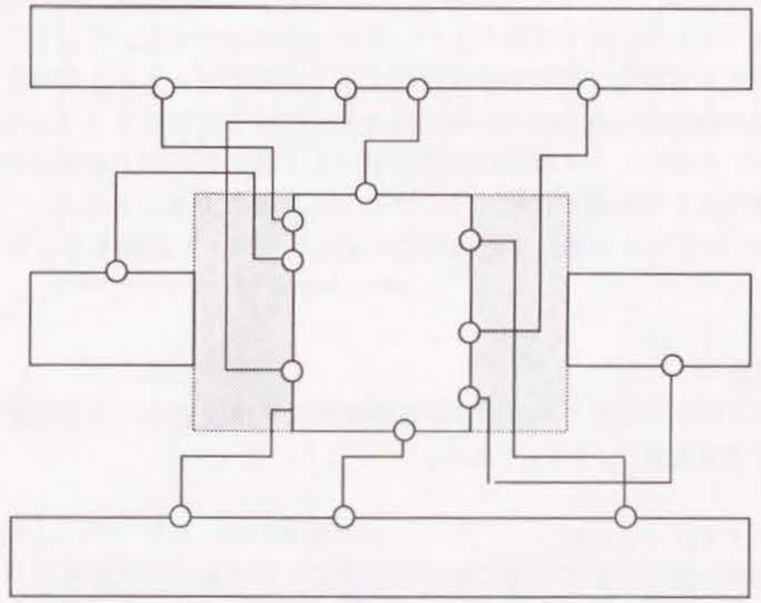


図4．8　チャネルの種類



（a）　上下辺に端子をもたせたマクロセル



（b）セル隙間チャネルを用いた従来の配線

図5．4　従来のマクロセルの配線

５．４　従来の配置配線手法

従来は、ＲＯＭ、ＲＡＭ、ＰＬＡ、ＡＬＵなどのマクロセルは、マクロセル１個をブロックとして扱っていた。標準セルとマクロセルを同時に配線するレイアウト方式としては、Sato et al.（５３）の発表がある。このレイアウトモデルは、前節５，３で述べた拡張ブロックと同じ形をしているが、配置は固定的である。マクロセルを配置できる位置は予め定義された「ベンチ」と呼ぶ位置に限定され、人手で配置される。更に、標準セル列の中にマクロセルを混在させることはできない。しかし、提案するレイアウトモデルと手法ではこのような制限はない。

従来より、比較的小さなマクロセルをセル列内に標準セルと隣接して配置配線する方法は次の２つがある。

（ａ）マクロセルを標準セルと同じモデルにして扱う簡易な方法である。図５．４(ａ)のようにマクロセルのサイズを大きくして、すべての端子をマクロセルの上辺または下辺に引き出しておく。しかし、この方法は実用的ではない。マクロセルを設計する時点では、マクロセルがどの位置に使われるかは不明であるので、上辺または下辺のいずれにひきだせばよいか決められない。従って、通常は近い辺に引き出すことになる。このように予め上辺下辺に引き出してあるマクロセルを配線すると、迂回した配線が多くなり、ブロックの面積を大きくする。

（ｂ）マクロセルと標準セルの隙間に縦方向のチャネルを設けて側面端子を配線する方法である。この方法は、ブロック間配線に使われている。隙間の縦方向チャネルの長さは、図５．４(ｂ)のようにサイズの大きいマクロセルに合わせる。そして、このチャネルの配線後に、チャネル端に仮の端子を置いてセル列間のチャネルを配線する。この方法は、マクロセルが配置された場所に応じて上辺または下辺に引き出せる点では、第１の方法より良い。しかし、セル隙間の縦方向チャネル端に仮の端子があるために、サイズの小さい標準セルとの配線が迂回して、ブロック面積を大きくする。

本方法は、（ｂ）の方法を改良して迂回配線を減らしブロック面積を小さくするのが特徴である。

５．５　セル配置

セル配置はサブブロック分割とサブブロック内配置より構成する。主な配置手法は、クラスタリング２次元配置法[３６]である。

５．５．１　サブブロック分割

先ず、ブロックの全論理をレイアウト用のサブブロックの集合に分割する。

サブブロック分割のためにつぎの２項目を指定する。

（１）各サブブロックの核となるＮ個のマクロセル。

（２）各サブブロック内のセル列段数。

なお、サブブロック内のマクロセルの位置は指定しない。マクロセルのサブブロック内の位置は自動配置する。Ｎ個の核となるセルを指定して、ブロックは（Ｎ＋１）個のサブブロックに分割する。

サブブロック分割法

（１）以下の（１）～（５）をＮ回（サブブロックの核となるマクロセルの数）繰り返す。

（２）論理の結合に従ってセルのクラスタ木を作成する。

（３）Schuler-Ulrich 法[５６]を用いて、クラスタ木をトップから分解する。その各分解レベルで、節点の一次元相対配置を決める。

この処理の際、核となるセルは強制的に一次元セル列の端に移しておく。即ち、クラスタ分解の時、左（右）のサブブロックの核となるセルは一次元配列の左（右）端に固定しておく。また、他のサブブロックの核となるマクロセルは一次元配列の反対端に固定する。この段階で、セル相互の論理結合を反映したセル並びができる。

（４）セルの一次元セル列の左（右）端から長さＬの位置で、セル列を２つに切断する（図５．５）。但し、Ｌはサブブロック内の全セル幅の和であり、 |核となるマクロセル幅| × |指定セル列段数| の長さである。

（５）切断で切り出された左（右）側にあるセルにサブブロック番号を付け、サブブロックを構成する。そして、番号の付いたセルを除く。残った核となるセルがあれば（２）に戻る。

（６）残ったセル集合に番号を付け、核セルのないサブブロックを構成する。

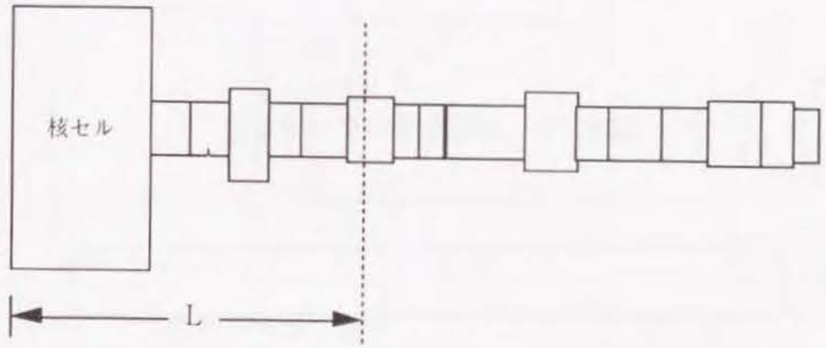


図５．５　サブブロック分割

５．５．２　サブブロック内配置

分割された各サブブロック内（各サブブロック番号の付いたセル集合）において、クラスタリング２次元配置し[３６]、ネットバランス配置改善を行う。[６７]この時左右のサブブロック内のセルとの結線も考慮する。

５．６　サブブロック内配線

５．６．１　左右の側面端子の配線

レイアウト面積を小さくするために側面端子に直接配線する。このため、先ず、マクロセルの側面端子を、図５．６に示す隣接セルの側面との相対関係により２つの属性に分ける。

（ａ）Ａ属性；端子のＹ座標が隣接セルの側面の範囲内にある。

（ｂ）Ｂ属性；端子のＹ座標が隣接セルの側面の範囲外にある。

明らかに各側面端子の属性は固定でなく、隣接セルの高さにより変化する。Ａ属性の側面端子を配線するには、セルとセルの隙間が必要である。Ｂ属性の側面端子を配線するには、セルの隙間は不必要である。Ｂ属性の端子は図５．７に示すように水平線により配線できる。従って、側面端子配線の目的はマクロセルセルと隣接セルの隙間を最小にすることである。

側面端子の配線は、セル列間チャネルへの引き出し配線とセル隙間配線の２方法に分けて処理する。

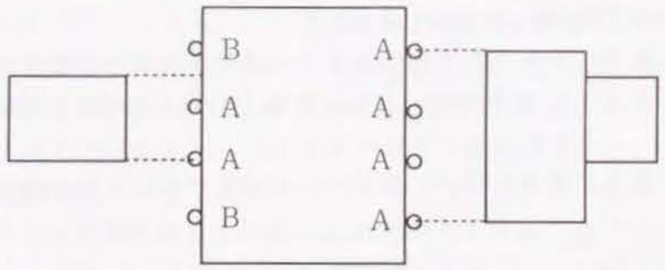


図５．６　側面端子の２つの属性

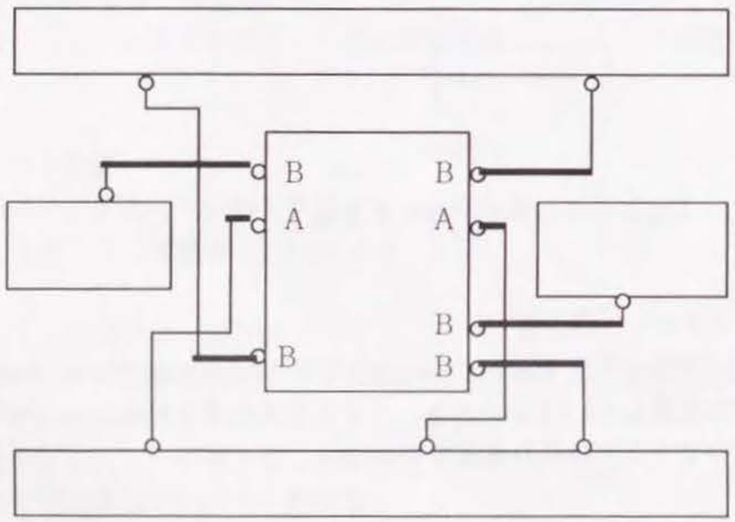


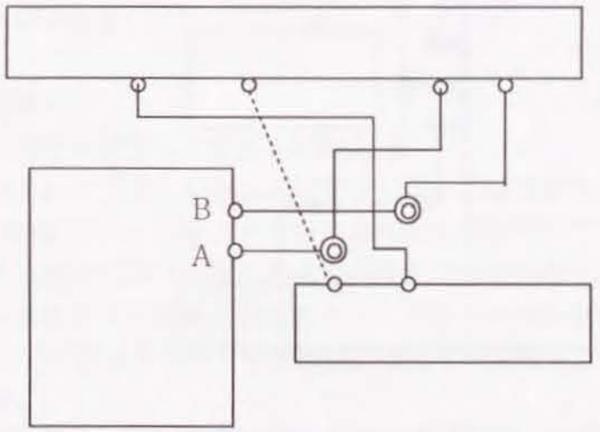
図５．７　側面端子からの水平配線

（１）セル列間チャネルへの引き出し配線

この方法を適用する端子は隣接セルの側面に同ネット端子がないＢ属性の端子である。この側面端子から水平線分を引き出し、線分の端に引き出し端子をおく。引き出し端子の位置は注意深く決めねばならない。なぜならば、図５．８に示すように他のセルの端子からの配線の障害になり得るからである。

引き出し配線法

側面端子のＹ座標位置に従ってセル列の上下辺上に引き出し端子の可能性を探索する。探索範囲は、側面端子から、障害となるセル側面かセル列の端までである。セル列間チャネルから選ぶ優先順序は、先ず、同ネットの最も近いセル上下辺の端子優先し、次いで、端子のないセル上下辺の内最も近い位置とする。もし、引き出し端子の位置が決まらない場合は、次節のセル隙間チャネル配線に委ねる。



◎　引き出し端子

図５．８　側面端子からの水平配線

（２）セル隙間配線

この方法を適用する端子は、Ａ属性の端子と前述の水平線分による引き出し配線できなかったＢ属性の端子である。セル間の隙間を小さくするために、セル列間チャネルにチャネル配線法を適用する。セル隙間チャネルは垂直方向である。幹線は垂直線で支線は水平線である。幹線の位置はチャネル配線結果で決まる。

セル隙間配線法

（１）各幹線を上に引き出すか下に引き出すかを決める。各幹線の上下方向を決める基準はネットの広がりに従う（図５．９）。即ち、ネットがブロックの上方（下方）に

広がるならば、引き出し方向はチャネルの上（下）部にする。ネットがブロックの上下両方に広がるならば、セル列間チャネルの混雑を減少させる方に決める。

（２）チャネル配線法によりトラック割付する。トラック割付によりセル隙間の幅が決まる。

（３）セル隙間の幅の分セルの座標をずらす。

（４）側面端子からの水平線である支線のみを残し、すべての幹線を消去する。水平線の端に仮端子（引き出し端子）を置く（図５．１０）。

引き出し端子は次節のセル列間チャネルにて配線する。

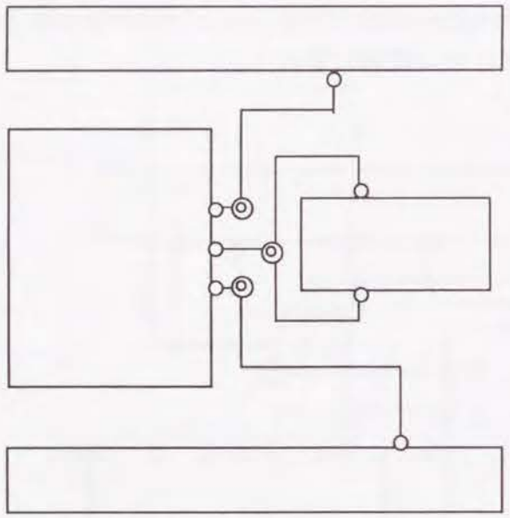


図５．９　幹線の方向

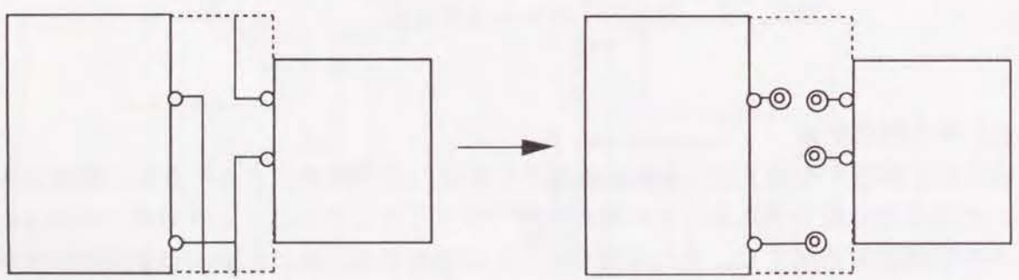


図５．１０　セル隙間チャネル配線

５．６．２　グローバル配線

本節では、スタイナ木を用いたグローバル配線、即ち、配線がセル列を渡る位置（フィードスルー）を決める処理を述べる。

フィードスルーの位置は２段階で決める。セル列と直交する垂直線を数十配線格子ごとに引き、水平線をセル列とセル列の間に引いて、この垂直水平線によりできる格子をマクロ格子と呼ぶ。フィードスルーは、先ずマクロ格子に割り付け、次いで、各マクロ格子内で詳細な位置を決める。マクロ格子を用いる理由は、配線処理順序の依存性を無くすためである。即ち、フィードスルーは中央部に集中する傾向があり、処理順に詳細位置を割り付けると後に処理するフィードスルーは適当な位置が割付済みになっていて迂回配線を生ずるからである。各マクロ格子ではフィードスルーを割り付けられる位置の数である容量を計数しておく。フィードスルーを割り付けられる位置とは、

（ａ）セル上、端子がなくて配線禁止でない位置、

（ｂ）セル上、フィードスルーの信号とと同じ信号の端子位置、

（ｃ）セル上、使用されない端子位置（上空を通るので短絡しない）、

（ｄ）セルのない位置、

である。

グローバル配線法

（１）マクロ格子を設定して容量を計数する。

（２）次の（３）～（４）をネット（信号）ごとに繰り返す。

（３）端子の位置を点としてスタイナ木を作成する。

（４）スタイナ木の枝とセル列と交差する位置のマクロ格子内に割り付ける。

（５）各マクロ格子ごとに割り付けたフィードスルーの数が容量を超えているか調べる。容量を超えているマクロ格子があれば、超えている分を近くで容量に余裕があるマクロ格子に移す。

（６）各マクロ格子内で、配線長が長くならないようにフィードスルーの詳細位置を決める。

グローバル配線の中での、複数のサブブロックに跨がるネットの扱いを述べる。複数のサブブロックに跨がるネットは、サブブロックの左側と右側に仮の端子（中継端子）を設ける（図５．１１）。但し、最左（最右）サブブロックの左（右）側の中継端子は、予めブロック外部端子として与えられている。中継端子は次のように決める。グローバル配線ではどのセル列間チャネルの左／右端にするかを決め、詳細なＹ座標は各サブブロック内の配置配線によって決定される。決定は左のサブブロックから右のサブブロックへ順に進める。

中継端子位置決定法

（１）ネットが左のサブブロックとつながるならば、左のサブブロックの固定した中継端子に最も近いセル列間チャネルの端に中継端子を割り当てる。

（２）ネットが右のサブブロックとつながるならば、このネットの最も右のセルの端子があるセル列間チャネルの右端に中継端子を決める。

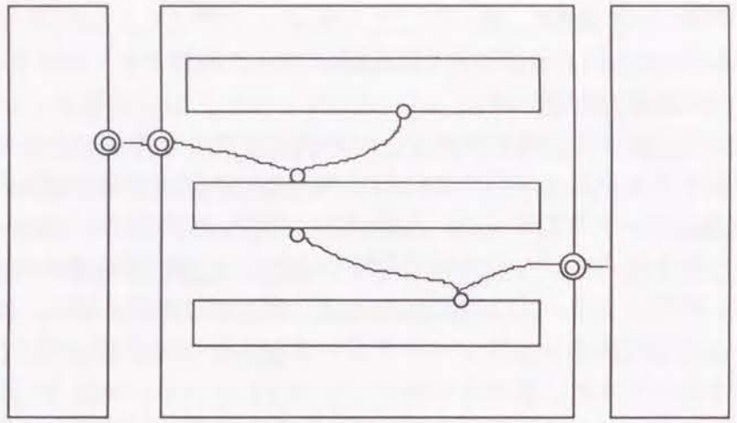


図５．１１　中継端子の位置の決定

５．６．３　セル列間チャネル配線

セル列間チャネル配線の対象となる端子は、セル上下辺の端子と、開放端子および閉塞端子の水平配線の引き出し点である。セル列間チャネルにはチャネル配線法を適用する。セル隙間チャネルとセル列間チャネルの重複領域にある側面端子の水平配線をセル列間チャネルの指定トラック幹線として扱えること、及び、マクロセルのサイズが標準セルよりも大きいことに対処できること、及び、種々のスルーホール隣接条件に対処できるために、従来のチャネル配線法に次の機能拡張をした。

（ａ）垂直制約グラフ

従来は、セルの上下辺の端子の間の垂直制約条件を用いて、垂直制約グラフを作成していた。これに対して、マクロセルの側面端子および側面端子からの引き出し点もセルの上下辺の端子と同様に扱って、これらの端子の間の垂直制約条件から垂直制約グラフを作成する。ただし、同ネットの端子間は垂直制約条件がないものとする。図５．１２は配線例とその垂直制約グラフである。垂直制約グラフにサイクルがなければトラック割付は幹線の屈曲なしに配線可能である。垂直制約グラフにサイクルがあればサイクルに従い幹線を分割屈曲して配線する。

（ｂ）トラック割付

トラック割付は１トラックごとに処理する。各トラックにおいて、側面端子からの水平配線は指定トラック配線として優先的に割り付ける。トラック割付は５．２．２節で作成した水平線（拡張配線とセル隙間配線）をトラック割付の時に固定幹線として扱う（図５．１２）。これらの線分は任意のトラックに割り付けることはできない。

トラック割付のトラック処理順序は原則として上下トラックを交互に中央に向かって処理する。ただし、単に上下交互を繰り返すと、サイズの大きいマクロセルの横が無駄領域になることがある。これを防ぐために、チャネル幅の予想値を、単線の重なり数と

チャネル内のセル突出部の高さの和の最大値で求めておく。そして、セル突出部の高さと反対側の配線済みトラック数の和がチャネル幅の予想値になった時、図５．１３のように、トラック処理順序を上下交互からセル突出部からの一方向に切り替える。これにより、サイズの大きいマクロセルの横に無駄領域が発生するのを防ぐ。

また、このトラック割付法の特徴の１つは、配線長に従って幹線のトラックを変更することである。トラック割付法は各トラックのみに注力するので、チャネル配線結果の配線長は常に最短とは言えない。図５．１４のように支線長を短くするために、垂直制約グラフの構造の制限の範囲内で幹線を上下方向に移動したり、交換したりする。

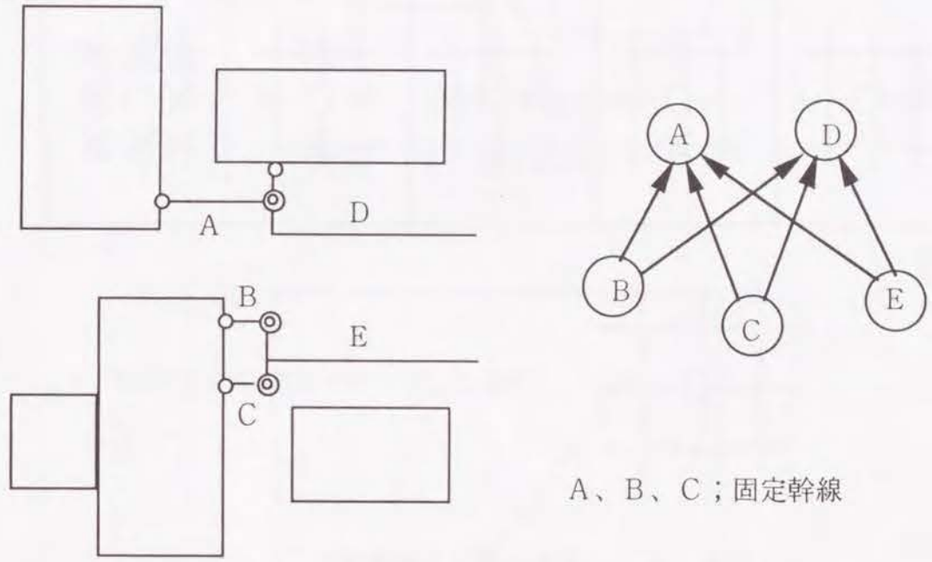


図５．１２　セル列間チャネル配線と制約グラフ

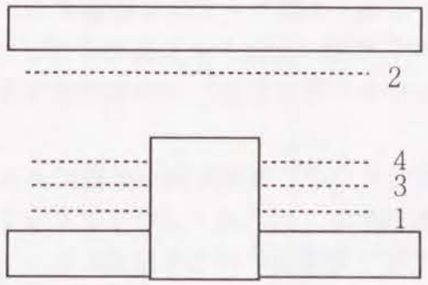


図５．１３　トラック処理順序

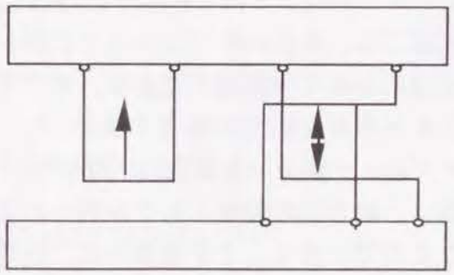
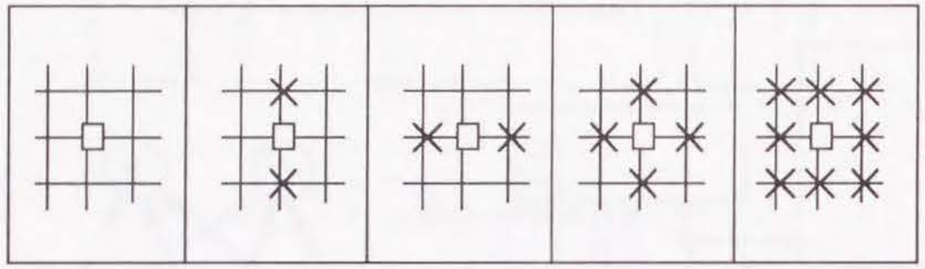


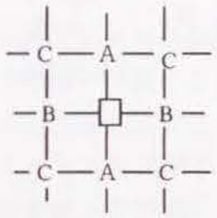
図５．１４　幹線の移動と交換

（ｃ）スルーホール隣接条件

プロセスによって配線とスルーホール相互の許容距離が異なる。配線講師の間隔を、

配線とスルーホールの許容距離にするか、スルーホールとスルーホールの許容距離にするかで、スルーホール隣接禁止の条件は図５．１５の５種類がある。これらのいずれに対しても配線可能にした。垂直方向の禁止（図５．１５のＡ点）は垂直制約グラフに反映させる。水平方向の禁止（図５．１５のＢ点）は各トラック内に配線する幹線相互の組合せを作る時にチェックする。斜め方向の禁止（図５．１５のＣ点）は、各幹線上のスルーホールと、前回のトラックに配線済みの幹線のスルーホールとの距離を調べることによりチェックする。





ＡＢＣは、スルーホール禁止設定可能位置

図５．１５　スルーホール隣接条件

５．７　サブブロック間配線

すべてのサブブロック内を配線した後に、各サブブロック間チャネルを配線する。対象となる端子は、複数のサブブロックに跨がるネットのセル列間チャネルでサブブロックの左右端に決めた中継端子である。サブブロック間チャネルには、セル列間チャネルと同じくチャネル配線法を適用する。

各サブブロック間チャネルには、サブブロックの間をつなぐ電源配線が必要である。電源配線は、縦方向の幹線と各セル列への支線で配線する。列の高さはサブブロックごとに異なるので、図５．１６のように、信号線の支線と電源線の同じＹ座標にあって、垂直制約条件をなすことがある。この垂直制約条件を満たすために、電源線もサブブロック間チャネルのチャネル配線法によりトラック座標を決める。

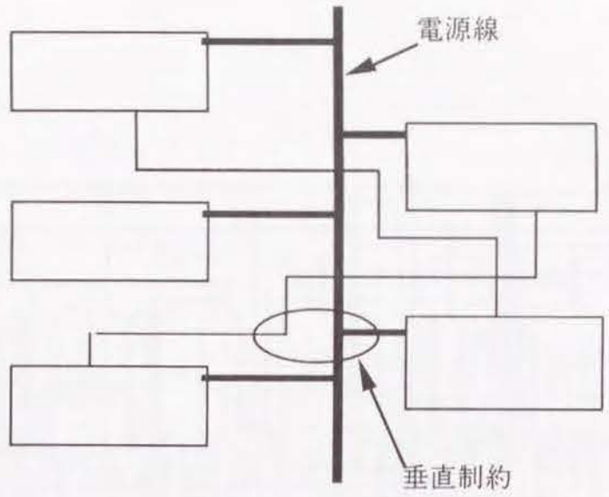


図５．１６　電源配線

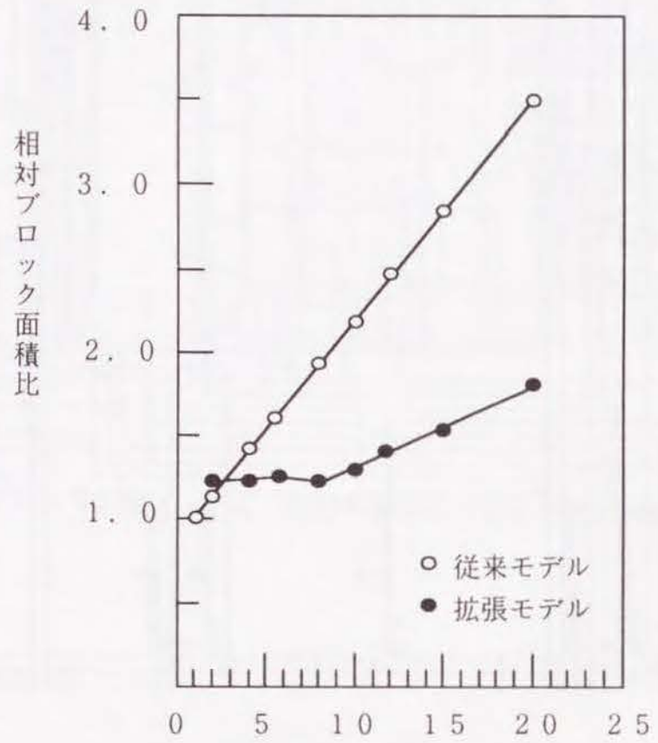


図５．１９　レイアウトモデルの比較

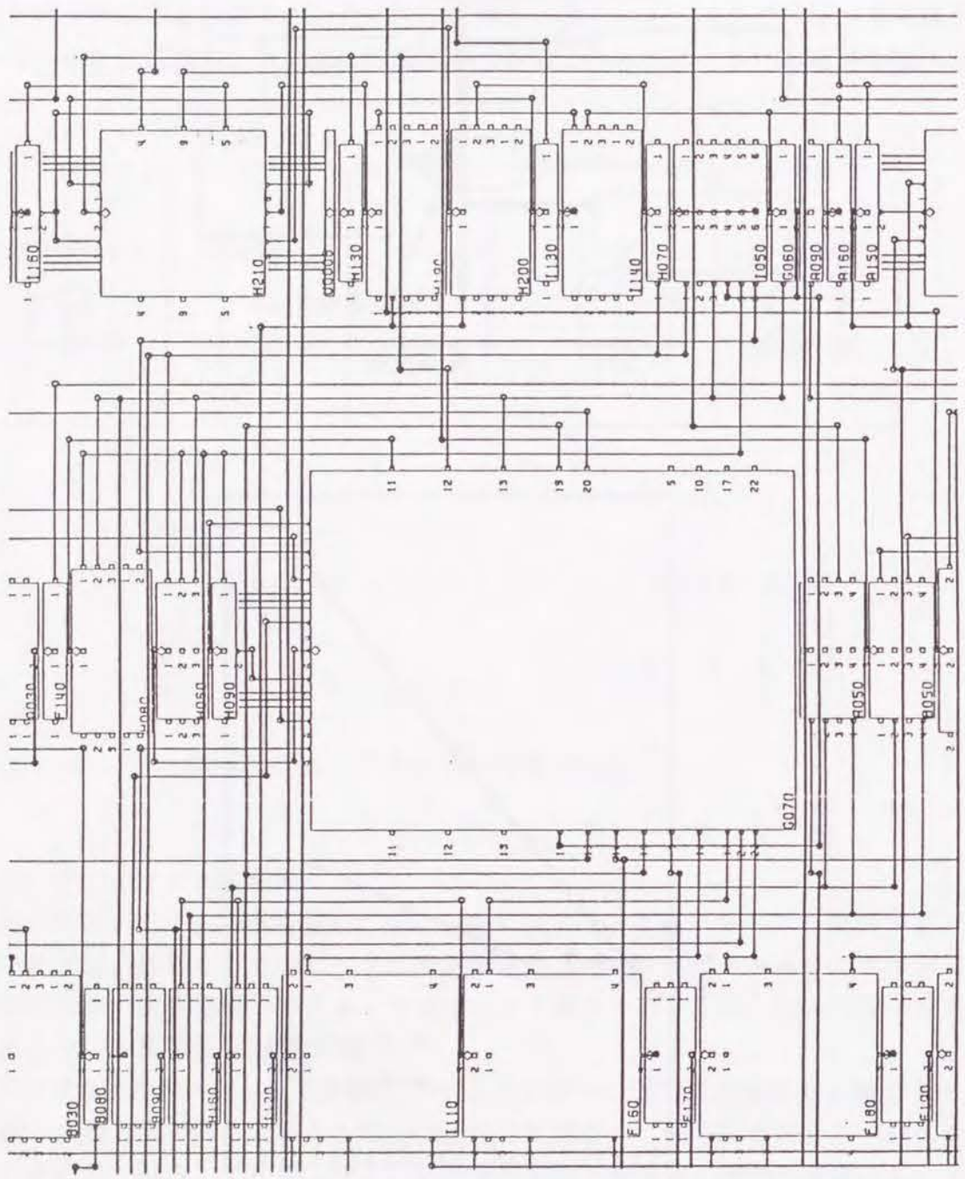


図５．１７　サブブロック内配線結果

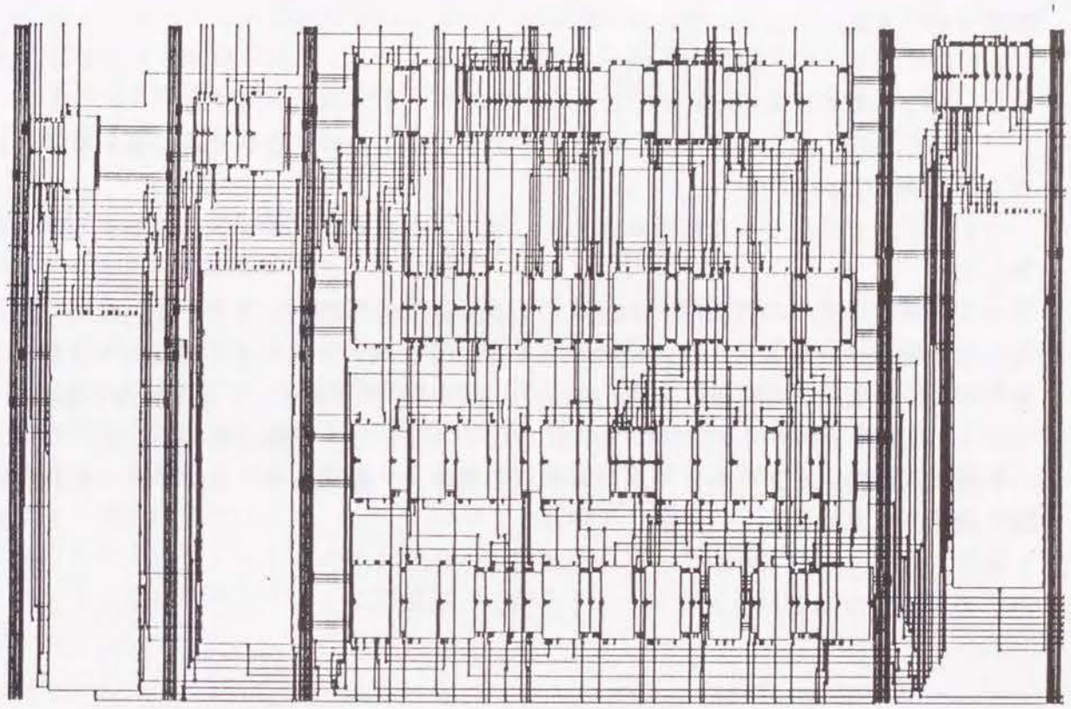


図５．１８　拡張モデルの自動レイアウト結果

５．８　結果と評価

サブブロック内の配線例を図５．１７に示し、拡張モデルの自動レイアウト結果を図５．１８に示す。セルの高さがブロック面積に及ぼす影響を、拡張モデルと従来モデルを比較することにより調べる実験をした。結果を図５．１９に示す。拡張モデルは、セルの高さが他のセルの３倍以上になると、従来モデルより優れていて、レイアウト面積を削減できることが明らかになった。

１５０セルで１９８ネットの場合、処理時間（Ｍ２８０Ｈ）は、配置に２秒、配線に９秒であった。

５．９　本章のまとめ

（１）サイズの大きく異なる、標準セルとマクロセルの混在させる拡張レイアウトモデルを提案した。

（２）巨大マクロセルを核セルとして、ブロックをサブブロックに分割する手法を開発した。

（３）マクロセルの側面端子の配線による無駄領域を最小に押さえる配線法を開発した。セル隙間チャネルとセル列間チャネル設けて、端子からの水平線はセル隙間チャネルのチャネル配線法で配線し、垂直線はセル列間チャネルのチャネル配線法で配線する。

（４）拡張モデルがこのマクロセルに有効であることを実験結果で示した。

本方法は自動レイアウトプログラムとして実現し、論理ＬＳＩのレイアウト設計に実際に適用されている。

第６章　ブロック間配線

６．１　はじめに

本章では、ブロック間配線の方法について述べる。ブロックをレイアウト設計して、フロアプランによりチップ内のブロック配置を決めた後に、ブロック間を配線する上位階層のレイアウトである。ビルディングブロック方式でのブロックの配置は、直線によって再帰的に２分できるスライス構造配置と再帰的には２分できない非スライス構造配置に分かれる。非スライス構造配置は従来良い配線方法がなかった。そこで、チップ面積最小化と１００％配線を目的として、従来のチャネル配線法を繰り返し用いることにより、非スライス構造配置を配線する「単純サイクル巡回配線法」を提案する[７８]。そして、評価結果を述べる。

６．２　ブロック間配線問題

（１）チャネル配線順序のサイクル問題

２．５．３で述べたように、ビルディングブロック方式でのブロックの配置は、直線によって再帰的に２分できるスライス構造配置と再帰的には２分できない非スライス構造配置に分かれる。スライス構造配置とは、図６．１のように、直線によって２分することを再帰的に行なうことができる配置である。一方、非スライス構造配置とは、図６．２のように、直線によって２分することを再帰的に行なうことができない配置である。非スライス構造配置では、２．５．２節の（３）の中で述べたように、チャネルの配線順序付けにサイクルが生じるので、チャネル配線法を用いて配線するのが困難であるというチャネル配線順序のサイクル問題[３３]があり、十年近く満足な解決法が見出されてなかった。

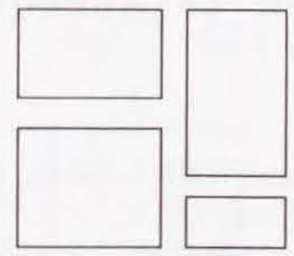


図６．１スライス構造配置

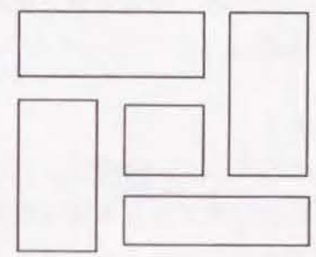


図６．２非スライス構造配置

チャネル配線順序のサイクル問題を再度説明する。チャネルとチャネルの接合状態によりチャネルの配線順序が決まる。これを配線順序グラフで表現する。配線順序グラフとは、図６．３のように、チャネルを頂点とし、チャネルＣ１とチャネルＣ２がＴ字形に接合している場合には、チャネルＣ１の頂点からチャネルＣ２の頂点へ有向枝を設ける。チャネルＣ１の配線結果によって、接合部の幅と接合部配線端の座標が決まる。こ

のチャネルの接合部での配線端を仮想的な端子とみなして中継端子と呼ぶ。チャネルＣ１の配線後であれば、チャネルＣ２は、対向２辺の端子位置がすべて決まって配線できる。従って、チャネルの配線順序は、配線順序グラフの有向枝の方向に従って行なう。ところが、配線順序グラフにサイクルがあるとチャネルの配線順序は決まらないので、配線不能である。これがチャネル配線順序のサイクル問題である。

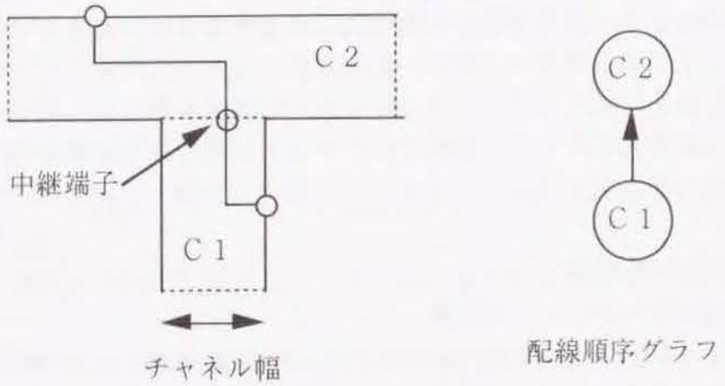


図６．３　チャネルのＴ字形接合

（２）電源配線

チャネル配線の長所は、配線に必要最小限な配線領域を確保して１００％配線することである。このため、信号配線をした後は電源配線をする余裕の配線領域はない。また、チャネル配線は線分の一次元配置を基本とする手法であるので、予め電源配線をしておいても電源配線を固定的に扱うことができない。さらに、電源配線は、信号配線と異なり配線幅が広い。このため、信号配線と同時に配線幅が広い電源配線をしなければならないという問題がある。

６．３　従来手法

チャネル配線順序のサイクル問題に対して多くの解決法が提案されてきた。しかし、ブロックの配置を移動してスライス構造配置に変更する方法[３５]は、ブロックを移動するために無駄領域を発生してチップサイズを増大する可能性がある。チャネルを細分してスイッチボックス配線する方法は、スイッチボックス配線が１００％配線できるとは限らないので未配線を生じて、１００％配線するチャネル配線の長所をなくしてしまう。Ｌ形チャネル配線[８][１１]は、Ｌ形チャネルの配線が少し無駄領域を発生する可能性をもっているが、今迄に提案された方法の中で最も良い方法と言える。

６．５　単純サイクル巡回配線法によるブロック間配線

ブロック間配線処理は、チャネル分割、グローバル配線処理、詳細配線処理からなる。

６．４．１　単純サイクル巡回配線の考え方

配線順序グラフのサイクルを、４つの頂点からなる単純サイクル（図６．４）と、複数のサイクルが組合さった複合サイクル（図６．５）の２種類に分類する。実際のレイアウトに現われるのは単純サイクルがほとんどである。本方法が対象とするのは、配線順序グラフのサイクル上のチャネルを巡回して配線できる単純サイクルである。複合サイクルがある場合には、ブロックを移動して複合サイクルを解消しておく必要がある。

図６．６に示す非スライス構造配置は、配線順序グラフ上で単純サイクルになっている。これを従来のチャネル配線で配線しようとすると、４つのチャネル接合部のいずれか、例えばＣ１とＣ４の接合部での、Ｃ４のチャネル幅と中継端子の座標を仮定し、Ｃ１、Ｃ２、Ｃ３、Ｃ４の順に配線する方法が考えられる。ところが、チャネルＣ４の従来のチャネル配線結果のチャネル幅と中継端子の座標は、最初仮定した値と一致しなければならない。一致する仮定値を設定するのは困難である。

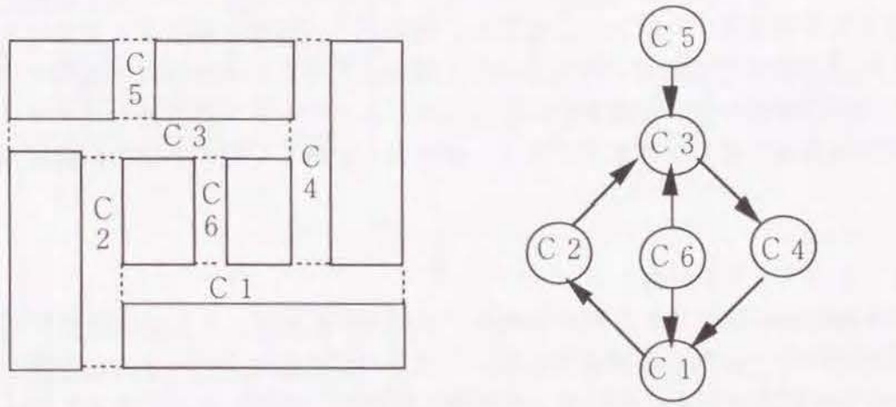


図６．４　単純サイクルを含むチャネルグラフと配置

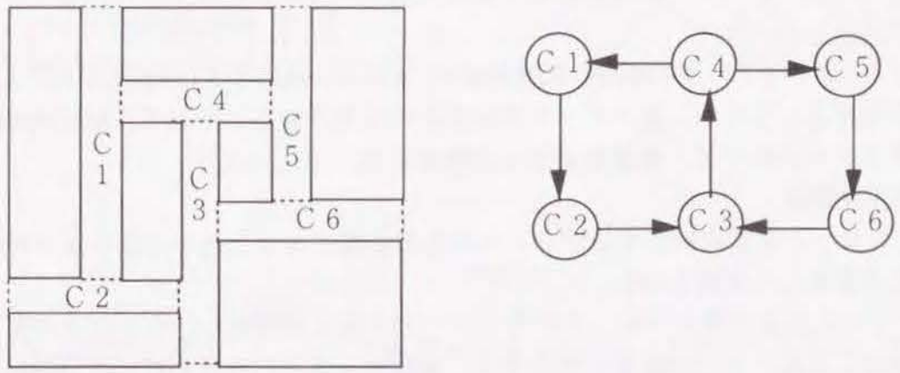


図６．５　複合サイクルを含むチャネルグラフと配置

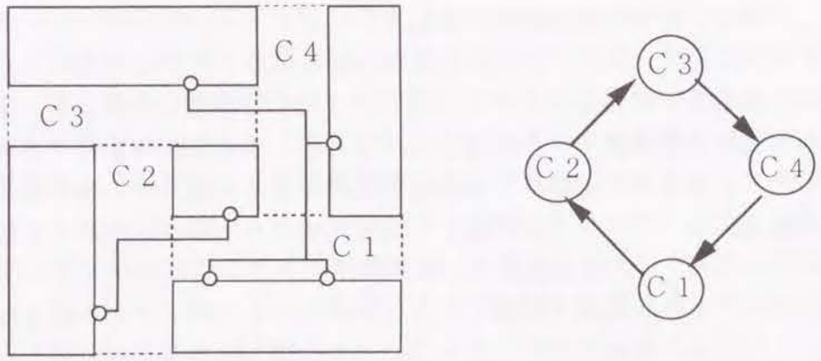


図６．６　単純サイクル巡回配線

そこで、１つのチャネル接合部のチャネル幅と中継端子の座標を適当に仮定し、サイクル内のチャネルを順次配線することを、チャネル接合部のチャネル幅と中継端子の座標が一致するまで継続すればよい。ただし、無限ループ処理に陥ることを防ぐために、繰り返し配線を途中で強制的に打ち切って、最後にチャネル接合部の中継端子の座標を固定した３辺固定チャネル配線をする。この時には、チャネル接合部でのチャネル幅と中継端子の座標は、適当な仮定値でなく、繰り返し配線した時の前回の配線結果の値を用いる。

６．４．２　チャネル分割

ブロック間配線に利用できる配線領域は、ブロックとブロックの間の隙間とブロック内の配線が利用せずに残った領域である。これらの配線領域をチャネルに分割する。

ブロック内の配線が利用せずに残った領域は既配線の隙間をぬって通過しなければならない。ここに配線を通す目的は、ブロックとブロックの間を少しでも小さくするためである。ブロック内を直線で通過できる場所を探し出して有限容量のチャネルとする。このチャネルは、ブロックの対向２辺と直角であるので配線順序グラフ上のサイクルの要因にはならない。

ブロックとブロックの間の隙間の配線領域を、Kimura et al.[３５]の方法を利用してチャネルに分割する。これは、後のチャネル割当法の対象である。そして、配線順序グラフに複合サイクルがあれば、複合サイクルを解消する。

チャネル分割法

（１）ブロック配置に対して、ブロックの各辺を他ブロック辺またはチップ外枠に到達するまで延長した直線を作る。

（２）２つの直線の間にブロックがなく、一方の直線の区間が他方の直線に含まれる包含関係にある時、２つの直線を併合する。直線が十字形に交差している場合は、２つのＴ字形の接合に分解する。すべてのブロック間に存在する直線が１本になるまで、この操作を繰り返す。できた直線に対応する配線領域をチャネルとする。

（３）配線順序グラフを作成する。

（４）配線順序グラフに複合サイクルがある場合には、複合サイクルを解消する。複合サイクルに含まれる各チャネルの両側に接合している２つのチャネルを探し、この２つのチャネルの間の距離が最も小さい組み合わせを求める。そして、この２つのチャネルを統合して、中間のチャネルを分割する。この操作を複合サイクルがなくなるまで行なう。

図６．７は、チャネルＣ１とＣ３を統合し、チャネルＣ２を分割することによって、複合サイクルを解消した例である。

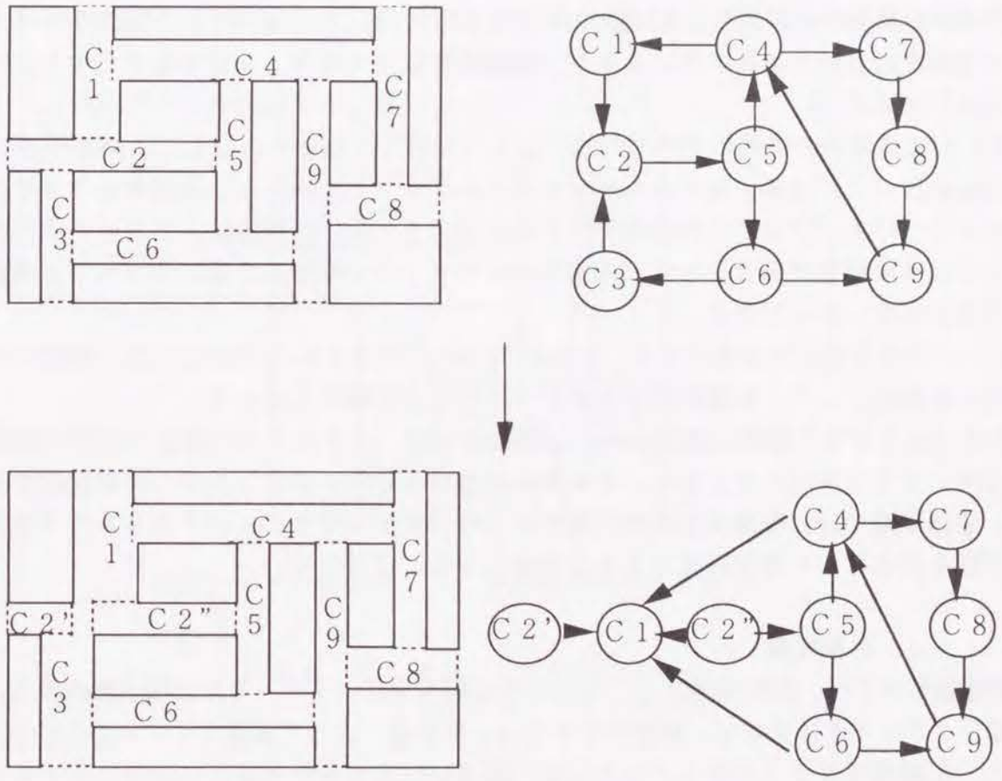


図６．７　複合サイクルの解消

６．４．３　グローバル配線

グローバル配線処理は、各ネットの配線が通過するブロック間のチャネル、及び、ブロック内の直線通過位置を決める。目的関数は、信号の配線長を短くすることである。配線方法は、グラフ探索型の三浦他[６３]の方法を利用する。従来にない機能として、バス信号のまとまった配線をする。

グローバル配線法

（1）チャネルグラフ（図2．8(b)）に、各ブロックに対してブロック内通過配線本数の制限値をもつチャネルを追加する。

（2）各ネットごとに、接続すべき端子座標の重心に最も近いチャネルを目標として設定する。2端子信号では一方の端子を目標にする。そして、各端子からこの目標のチャネルに向かって最短の配線経路をチャネルグラフ上で探索する。

バス信号は、複数の信号から構成されて纏まった信号であるので、信号相互のスキュー値を小さくしなければならない。スキュー値を小さくするためには、バス信号内の信号の配線長をそろえる必要がある。従来は、バス信号内の複数の信号を独立に配線していたので、配線長をそろえることは難しかった。このため、バス信号内の配線を同一チャネルを通る配線経路にすることによって、配線長を揃えてスキュー値を小さくする。

バス信号配線法

（1）バス信号内の信号のすべての端子を1つの信号の端子とみなして、前述のグローバル配線法により通過するチャネルをチャネルグラフ上で決める。通過するチャネルにマークをつける。ブロック内部のチャネルは通過させない。理由は、ブロック内部のチャネルは有限容量であるためにバス信号内のすべての信号が必ず同じチャネルを通過できる保証がないからである。

（2）バス信号内の各信号ごとに、マークのついたチャネル内に限定して、前述のグローバル配線法により、配線経路をチャネルグラフ上で探索し決める。

本方法によりできる配線経路にはループがないので、バス信号内の複数の信号の配線経路は同じチャネル内にまとまる。チャネル内部のトラック位置は後述の詳細配線で決める。詳細配線では、幹線の垂直制約関係により、必ずしもすべてのチャネル内で信号の並び順を同じにした定常流線のような束線になるとは限らない。

6．4．4　詳細配線

詳細配線処理では、配線順序グラフの有向枝の順序を守ってチャネルを順次配線する。配線順序グラフのサイクルは、前述のチャネル分割処理によって単純サイクルのみになっている。配線順序グラフのサイクル上にない頂点に対応するチャネルは従来のチャネル配線する。

単純サイクル巡回配線法

（1）サイクル内の任意の1つのチャネル接合部のチャネル幅をグローバル配線で求まった配線重複数に設定し、中継端子の座標を配線長最小になる位置に仮定する。

（2）4つのチャネルを順に従来のチャネル配線する。そして、各チャネル接合部のチャネル幅と中継端子の座標を、前回値として記憶する。

（3）4つのチャネルの配線を終えた後は、1つのチャネルを配線するごとに、チャネル幅と中継端子の座標がすべて前回値と一致するか比較する。

（4）すべて一致すれば、サイクルをなす4つのチャネルの配線を終わる。

不一致の値があれば、今回の配線結果のチャネル幅と中継端子の座標値を前回値として記憶し、同時に、不一致数も記憶する。そして、次のサイクル内のチャネルに移る。

（5）4つのチャネルの配線処理の繰り返し回数が規定値を超えた場合には、強制的に繰り返し配線を打ち切って、4つのチャネル接合部の中で、チャネル幅と中継端子の座標値の不一致数が最小のチャネル接合部を選ぶ。そして、この接合部の次のチャネルから順に3つのチャネルを従来のチャネル配線する。そして、4つ目のチャネルは、チャネル接合部のチャネル幅と中継端子の座標を前回値に固定して、3辺固定チャネル配線する。

図6．8を例に説明すると、チャネルC1とC2の間の接合部が不一致数最小であったとすると、チャネルC1とC2の接合部のチャネル幅と中継端子の座標を前回値に固定して、チャネルC2、C3、C4を従来のチャネル配線する。その後、チャネルC1を3辺固定チャネル配線する。

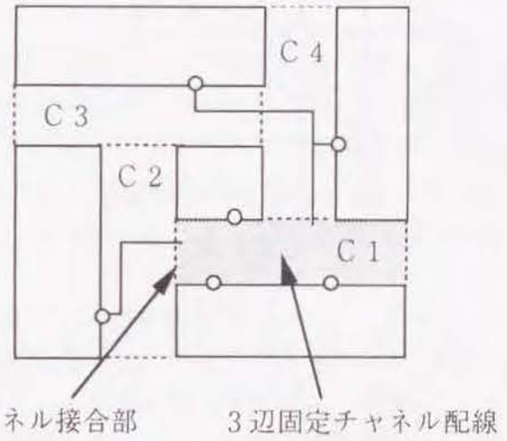


図6．8　繰り返し強制打切り後の
3辺固定チャネル配線

3辺固定チャネル配線法

3辺固定チャネル配線とは、対向2辺の端子位置に加えて、もう1辺の中継端子の座標とチャネル幅を固定して配線する配線法である。3辺固定チャネル配線では、チャネル幅と一方の中継端子の座標が固定されるので、未配線を発生する可能性をもっている。そこで、未配線の可能性を小さくするために次のように工夫した。

（1）対向2辺の端子による垂直制約条件を抽出して、垂直制約グラフを作成する。更に、中継端子の座標の並び順を垂直制約条件として垂直制約グラフに追加する。

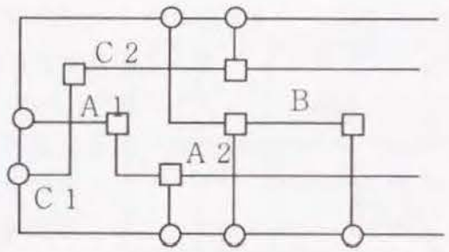
（2）垂直制約グラフサイクルがあれば、幹線を分割してサイクルを解消する。この時、中継端子につながる幹線に対しては、対向2辺の端子による垂直制約位置と中継端

子との中間位置で幹線分割する。

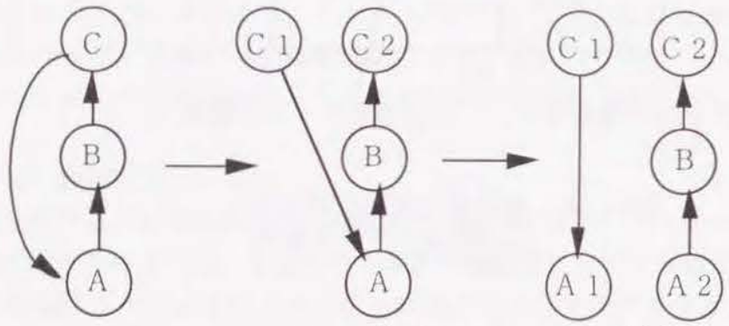
（3）中継端子とつながる幹線が制約パス上にあり、制約パスから決まる幹線の配線可能範囲にこの中継端子の座標がない時も、この幹線を対向2辺の端子による垂直制約位置と中継端子との中間位置で幹線分割する。

（4）幹線をトラック割付する。

図6．9に3辺固定チャネル配線における幹線分割による垂直制約グラフの変化を示す。図6．9(b)は、中継端子の並び順を垂直制約条件として追加した垂直制約グラフである。図6．9(c)は、サイクルを解消するために、幹線CをC1とC2に分割した後の制約グラフである。図6．9(d)は、制約パスにより幹線の座標が中継端子の固定座標に合わないので、幹線AをA1とA2に分割した後の垂直制約グラフである。この配線結果が図6．9(a)である。



（a）配線結果



（b）　（c）　（d）

垂直制約グラフの変遷

図6．9　3辺固定チャネル配線

6．5　電源配線

電源配線は信号線と同時に配線する。電源配線の幅広幹線が占める場所以外は他の信

号配線が有効利用できるように、幅広幹線である電源配線と普通幹線（1トラックのみ占める幹線）である信号配線を混在して同時にトラック割付する。幅広幹線には配線幅を配線格子の幅で割り算した値を連続割付トラック数として属性を付けておく。

幅広幹線のトラック割付法

（1）幅広幹線と普通幹線を含めた垂直制約関係を抽出して、垂直制約グラフを作成する。

（2）垂直制約グラフ内の制約サイクルの有無を調べる。幅広幹線と普通幹線を含んだ制約サイクルがある場合には、普通幹線を屈曲配線（幹線分割）することにより制約サイクルを除去する。

（3）トラックごとに幹線の割り付けを繰り返す。この時、幅広幹線に関して(a)～(e)の処理を行う。

(a) 幅広幹線が初めてトラック割付されたら、連続割付トラック数を記憶する。

(b) トラック割付途中の幅広幹線は優先的にトラックに割り付け、連続割付トラック数を－1する。

(c) トラックの処理順序は原則チャネルの上側下側を交互に中央に向かう。しかし、幅広幹線のトラック割付途中はトラックの処理順序を上側または下側から中央に向かう一方向に固定する。これは幅広幹線が上下に分離しないためである。

(d) 幅広幹線と重ならない普通幹線があれば、同じトラックに割り付ける。

(e) 連続割付トラック数が0になったら幅広幹線は割付け済みである。

6．6　ボンディングパッドの配置配線

チップレイアウトが終わらないとチップサイズが決まらない。チップサイズが決まらないとボンディングパッドの配置が決め難いという問題があった。従来は、予想チップサイズに基づいて適当なサイズに作り、内部の論理領域のレイアウト後に詳細なボンディングパッドの位置を決定していた。そこで、この機能を自動化して、レイアウトしながらボンディングパッドの位置を決定する機能を実現した。このために、ボンディングパッドと入出力バッファをセルにして、ブロック（ボンディングパッドブロック）の中に指定してもらう。

ボンディングパッドの配置配線法、

（1）内部の論理領域の配置配線後、ボンディングパッドブロックの幅を内部の論理領域の幅高さに合わせて変更する。

（2）ボンディングパッドブロック内のセルを、ボンディングパッドブロック幅に合わせて分散配置する。

（3）ボンディングパッドブロックと内部の論理領域の間のチャネルをチャネル配線する。

（4）ボンディングパッドブロック内のセル側面の電源端子間を配線して、周回電源配線にする。

６．７　実験結果

図６．１０と図６．１１は、非スライス構造配置のチャネル構造とブロック間配線結果を示す。ブロック数は１６、ネット数は２５０、端子数は５２１である。この非スライス構造配置におけるサイクル内の４つのチャネルの配線は、７回目のチャネル、即ち、繰り返し２周目にチャネル幅と中継端子座標がすべて一致して終了した。繰り返し配線のみで十分であり、３辺固定チャネル配線は不要であった。配線処理時間は１６．８秒（Ｍ－６８０Ｈ）であった。

６．８　本章のまとめ

（１）非スライス構造配置に対して、チャネルを巡回して繰り返し配線する「単純サイクル巡回配線法」を提案した。本方法は、配線順序グラフの単純サイクルに対応するチャネルを繰り返し配線することにより、チャネル幅と、中継端子の座標を一致させる方法である。また、繰り返し回数を抑さえるための３辺固定チャネル配線法についても述べた。

（２）「単純サイクル巡回配線法」の実験結果は、繰り返し配線のみで十分であり３辺固定チャネル配線は不要であった。これにより本方法が有効であることを確認した。

（３）チャネル配線法を拡張して、信号配線と同時に配線する電源線を幅広配線する方法を開発した。

本方法を組み込んだブロック間配線プログラムは、製品の論理ＬＳＩのレイアウトに使用されている。

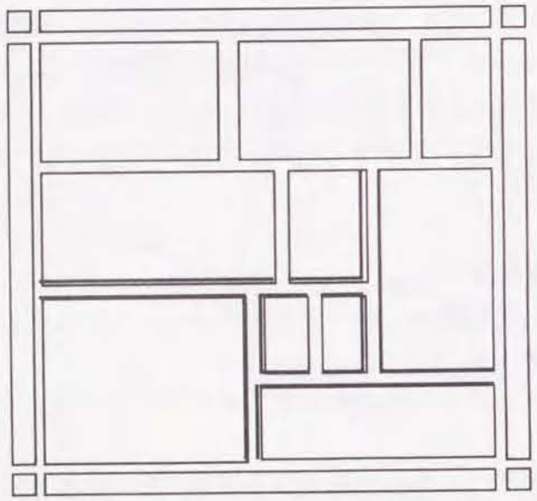


図６．１０　ブロック間配線結果のチャネル構造

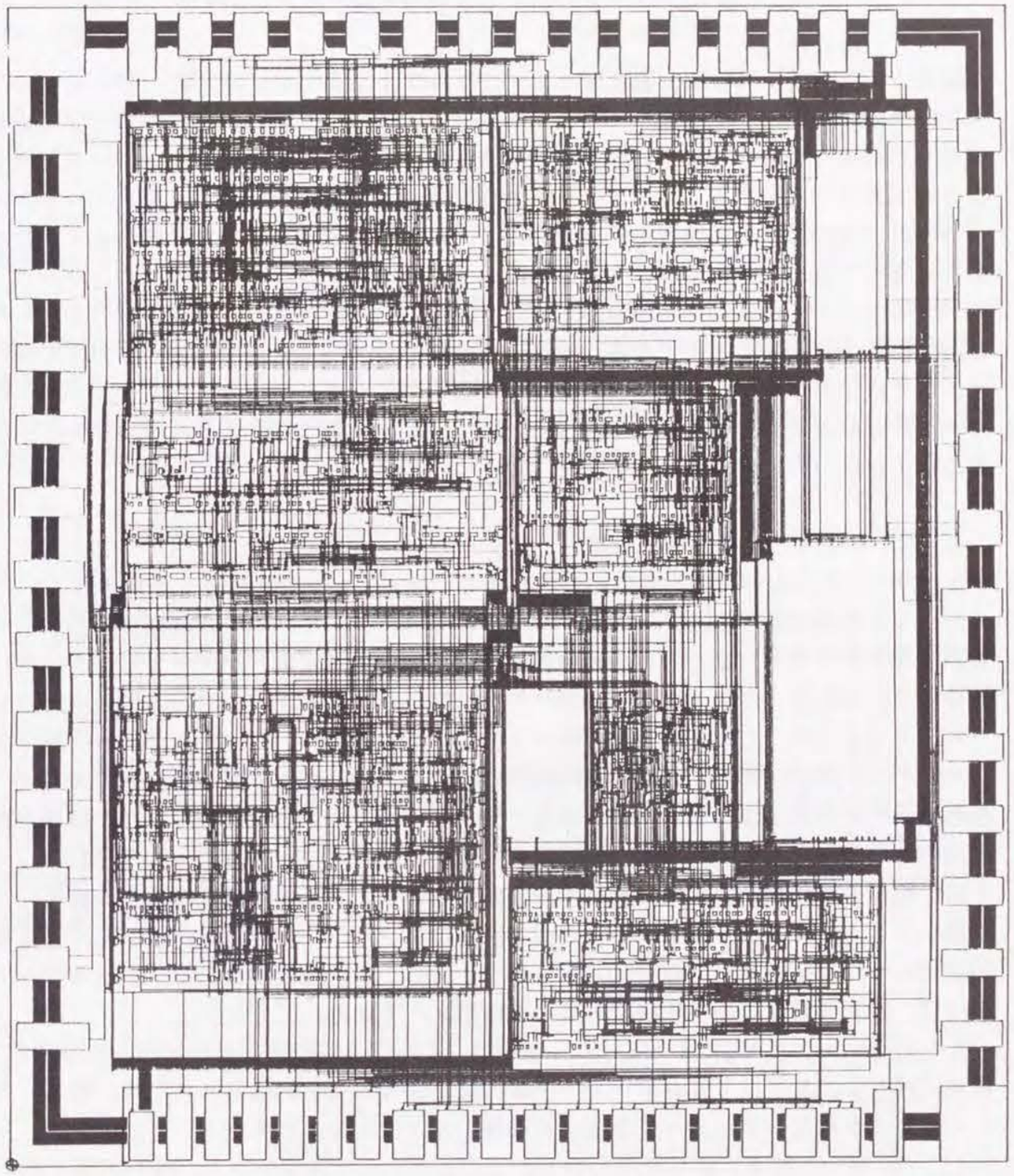


図６．１１　非スライス構造配置のブロック間配線結果

第７章　知識の記憶最小表現

７．１　はじめに

論理ＬＳＩには、第５章、第６章で述べたビルディングブロック方式とゲートアレー方式がある。ゲートアレーはチップサイズが予め決まっているので、配線量により配線領域を増減することができず、配線結果に未配線が発生することがある。この未配線を扱うルールベースのエキスパートシステムが発表されている[１６]。エキスパートシステムでは大量の知識を記憶する必要があるため、大量の知識を効率良く記憶する手法が必要となって来ている。実際に利用されているエキスパートシステムはプロダクションルールのシステムが多く、命題論理で表現できるものが多くある。命題論理の各ルールは論理関数で表わすことができる。本章では、ルールベースを論理関数の積和形で表現し、二分決定グラフを用いて積和形を高速に最小化することによって、ルールベースのルール数を削減する手法について述べる。特に、ホーン節に対応する論理関数の巡回成分の極小化について述べる。

７．２　等価性と非冗長積和論理

ルールベースのルールは含意関係であるので、通常、ｘｙ→ｚの形で表わされる。このルールを論理関数で表わすと、$\overline{x}+\overline{y}+z$となり、ルールベースは和積標準形として表わすことができる。また、この和積標準形は、否定を考えるとド・モルガンの法則により、積和標準形となる。各ルールが、左辺は正リテラルのみで、右辺には一つのリテラルのみである時、このルールは「ホーン節」であると言う。各ルールがホーン節であるという条件を仮定すれば、和積標準形では各和項には高々一つの正リテラルしかなく、積和標準形では各積項には高々一つの負リテラルしかない。論理関数は、どの積項も積項内のどの変数も除くことができない時、非冗長であると言う。また、論理関数のコストは、積項数と積項内の変数の和との加重和で表わす。同じ論理関数の種々の表現の中で最小コストである時、最小であると言う。一般に、一つの論理関数に対して非冗長積和論理は一つとは限らず、最小形とは限らない。しかし、論理関数が正リテラルのみであれば、非冗長表現は最小表現であることが知られている。

$X=x_1x_2\cdots x_n$、$Y=y_1y_2\cdots y_m$という２つの正リテラルのみから成る積項に対して、$X\overline{Y}+Y\overline{X}$がルールベースの論理関数の否定に包含される時、即ち、ルールベースからＸ→ＹとＹ→Ｘが導かれる時、ＸとＹは等価であるという。

ここで、ホーン節を、正リテラルのみから成る正リテラル成分Ｐと、唯一の負リテラルを含む純ホーン成分Ｈに分類する。ホーン関数ｆは、ｆ＝Ｐ＋Ｈと表わすことができる。これに関して、次の定理[２０]が知られている。

[定理] ホーン節集合に対応する非冗長論理関数をｆ＝Ｐ＋Ｈとすると、

（１）Ｐは最小である。

（２）次の２条件を満たす時Ｈは最小である。

(a) 等価でないP_iとP_jを正リテラル部にもつ主項の対がない。

(b) 正リテラル部P_iをもつ主項に対して、$P_i(\overline{X}+\overline{Y})\rightarrow f$なる、等価な正積ＸとＹの対がない。

この定理は、純ホーン成分Ｈの最小化のみを考えればよいことを示している。

７．３　二分決定グラフによる非冗長積和論理の生成

０抑制二分決定グラフ（ＺＢＤＤ、Zero-suppressed Binary Decision Diagram）は、論理関数をコンパクトに表現し、入力変数順を固定することにより関数を一意に表すことができる。ＺＢＤＤを用いた非冗長積和論理は、湊により発表されたＩＳＯＰ（Irredundant Sum Of Products）アルゴリズム[６５]により高速に生成できる。等価な正リテラル部が存在しない場合は、このアルゴリズムを利用して高速に最小化できる。しかし、等価な正リテラル部が存在する場合は、必ずしも非冗長積和論理が最小にはならない。従って、次節以降の処理をする。

７．４　巡回含意法による極小化

７．４．１　等価な正リテラル部の検出

非冗長積和論理表現に等価な正リテラル部が存在する場合は、必ずしも積項数が最小になるとは限らない。そこで、一旦、非冗長積和論理で表現した後、更にこれを簡単化する。

先ず、非冗長積和論理によって得られる積項に同じ正リテラル部があればこれでくくる。そして、これを２列の表形式で記憶する。この表を「積項表」と呼ぶ。列の値は正リテラル部と負リテラル部を表わす。先頭列の値が同じ行はマージする。この時、２列目の値はマージ前の２列目の値の論理和にする。この表に対応する論理関数はＺＢＤＤで表現する。表７．１に積項表の例を示す。

表７．１　積項の例

正リテラル部	負リテラル部
$x_1x_2x_5$	$\overline{x_4}$
x_2x_4	$\overline{x_1}$
x_3	$\overline{x_2}+\overline{x_5}$

表が表わす関数は$f=x_1x_2x_5\overline{x_4}+x_2x_4\overline{x_1}+x_3\overline{x_2}+x_3\overline{x_5}$。

次に、等価な正リテラル部の検出をする。積項表のすべての正リテラル部に対して、等価な正リテラル部を全て検出する。この時複数の等価な正リテラル部が存在すれば、それらを「等価クラス」と呼ぶ。

等価な正リテラル部の検出手順；

（1）$\overline{f_0}$の正リテラル部のみからなる主項（prime implicant）P_iを求める。
（2）$\overline{P_i x_e}$がｆを包含すれば、このP_iを等価クラスの要素とする。
（3）正リテラルのみからなる主項がなくなるまで（1）（2）を繰り返す。

ただし、ｆはルールベースが表わす論理関数の否定、Ｐは検索する正リテラル部、x_eはテンポラリー変数（ｆに現われない変数）、$f' = f + P_i \overline{x_e} + \overline{P_i} x_e$、$f_0$は$f'$に$x_e = 0$を代入した関数である。
このようにして、要素が複数のものを等価クラスとする。

７．４．２　含意関係の書き換え

このようにすべての等価クラスを検出した後、それぞれについて積項表を書き換えて、即ち、含意関係を書き換えて、最小化する。
ホーン節の含意関係を表わすのにグラフを定義する。点は変数または変数の積とする。含意関係がＸ→Ｙならば、点Ｘから点Ｙへの有向枝をつける。ＸとＹが等価ならば、ＸとＹはグラフの強連結成分に含まれる。提案手法のアイデアを図７．１に示す。ＩＳＯＰ法で作られたホーン節の含意関係を表した図７．１（ａ）を図７．１（ｂ）のように書き換える。この書き換えは、次のステップＳ１、Ｓ２、Ｓ３で行う。

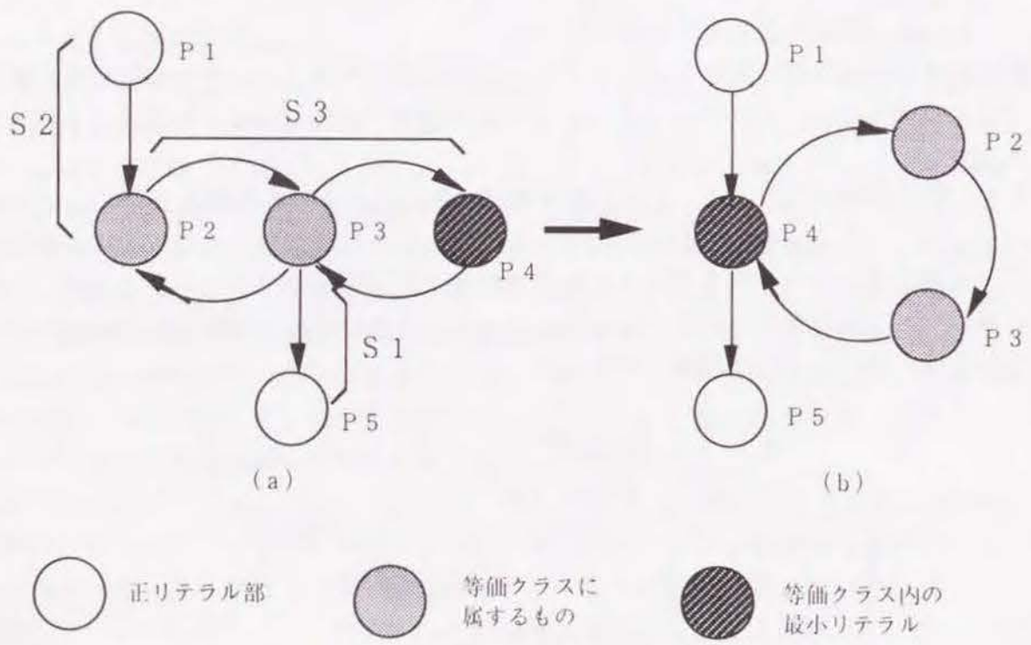


図７．１　含意関係の書き換え

ここで以下のように定義する。
$G_k = (g_{k1}, g_{k2}, \cdots, g_{knk})$は等価クラス。$g_{ki}$（$1 \leqq i \leqq n_k$）は正リテラル部。$g_{min(k)}$はリテラル数最小の$G_k$の要素（複数個存在する時はいずれか一つ）。$V_k$は$G_k$の要素に含まれる正リテラルの集合。$x_i$は正リテラル。$t_j$は積項表に存在する正リテラル部とする。

Ｓ１；　出力枝の変更（等価クラス外への含意関係の書き換え）
積項表の正リテラル部の行にg_{kl}があり、負リテラル部の行に$\overline{x_j}$（$x_j \in V_k$）が存在する時、積$g_{kl}\overline{x_j}$を$g_{min(k)}\overline{x_j}$に書き換える。これによりリテラル数を削減する。図７．１では枝の出発元をP_3からP_4へ換える。

Ｓ２；　入力枝の変更（等価クラス外からの含意関係の書き換え）
積項表の負リテラル部の行にt_j（$t_j \in G_k$）と$\overline{g_{kl}}$が存在する時、$\overline{g_{kl}}$を$\overline{g_{min(k)}}$に書き換えて、$t_j \overline{g_{min(k)}}$とする。これによりリテラル数を削減する。図７．１では枝の到達先をP_2からP_4へ換える。

Ｓ３；　枝の削減（等価クラス内からの含意関係の書き換え）
図７．１のように含意関係の２つのループを一つのループにする。また、他の含意関係から部分的に導出できるものも考える。図７．１では２つのサイクルを一つのサイクルにする。

Ｓ３－１；　行列の作成
先ず、積項表に$g_{kl}\overline{x_i}$（$x_i \in V_k$）をみたすx_iが存在すれば、これをすべて削除して積項表を縮小する。次いで、削除により等価性が失われるので適当な積項を加えることによって最初の関数と論理的に等価にする。
適当な積項を加えるために、$n_k \times n_k$の行列Ａ、Ｂを作る。行列Ａのｉ行ｊ列の値Ａ（ｉ、ｊ）は$g_{ki} \to g_{kj}$を遷移的表現にするために、縮小した積項表内の積項以外で必要となる積項数を表す。$g_{kj} = x_{j1} x_{j2} \cdots x_{ju}$の場合、$g_{ki}\overline{x_{j1}}$、$g_{ki}\overline{x_{j2}}$、・・・、$g_{ki}\overline{x_{juの}}$ようなｕ個の積項を加えてもよい。しかし、$g_{ki}$と$g_{kj}$に同じリテラル$x_l$が含まれる場合は、そのリテラルへの含意$g_{ki}\overline{x_l}$は不要である。また、縮小した積項表内に$t_v\overline{x_j}$（$t_v \supset g_{ki}$、$x_j \in \{x_{j1}, x_{j2}, \cdots, x_{ju}\}$）があれば、積項$g_{ki}\overline{x_j}$は不要である。また、行列Ｂは必要となる積項の負リテラルの論理和を表す。ただし、Ａ（ｉ、ｊ）＝０の時Ｂ（ｉ、ｊ）＝真とする。

Ｓ３－２；　極小表現
このようにして作成した行列を用いて、加える積項数を最小にしながらＧｋに属する任意の正リテラル部g_p、g_qに対して等価性が保持される。即ち、$g_p \to g_q$が導かれるようにする。一般的には、積項を加えるごとに、これらの行列の値を書き換え、等価性判定を行ない、最小となる必要数をもとめなければならない。そこで、最小解ではないが近似解を求める。即ち、ループ状に$g_{k1} \to g_{k2} \to g_{k3} \to \cdots g_{k1}$と表現されるように積項を加える。これにより、行列の書き換えの必要がない。以後は、この近似解

を極小解と呼ぶ。

Ｓ３－３；　並び順の決定

次に、極小解となるループの並び順を求める。

（１）ｓ＝１、２、・・・、n_kに対して次のａ～ｆをする。

ａ）$N_s=0$、ｉ＝ｓ

ｂ）ｓ行を行列Ａ、Ｂから除く。

ｃ）順列P_sの最初の要素をｓとする。

ｄ）行列Ａのｉ行の最小要素を探して、この列をｊとする。

ｅ）$N_s=N_s+A$（ｉ、ｊ）

ｆ）P_sの後尾にｊを加える。

ｇ）Ａ（ｉ、ｋ）＝０、且つ、Ａ（ｋ、ｌ）＝０を満たすＡ（ｉ、ｋ）が存在すれば、ｋ行ｋ列を除く。

ｈ）行列Ａのｉ行ｊ列を除いて、ｉ＝ｊとする。

ｉ）行列ＡとＢが行も列もなくなればｊへ、そうでなければｄへ。

ｊ）$N_s=N_s+A$（ｉ、ｓ）

（２）すべてのN_sの中で最小値N_{smin}を求め、$s=s_{min}$とする。

（３）$i=s_{min}$を初期値として（１）の手順で順列P_{smin}を求める。

行列Ｂを用いて、P_{smin}の順序に従い積項を積項表に加える。これにより、クラスの等価性を保持して、積項数を削減する。

例えば、

$f=x_1x_2\overline{x_3}+x_1x_2\overline{x_4}+x_3x_4\overline{x_1}+x_3x_4\overline{x_2}+x_3x_1\overline{x_5}+x_5\overline{x_3}+x_5\overline{x_4}$

とする。x_1x_2とx_3x_4とx_5が等価であるので、ｆに対するグラフは図７．２(ａ)になり、積項表は図７．２(ｂ)になる。これの等価クラスの要素数は３であるから、２つの３×３行列ＡとＢは図７．２(ｃ)と(ｄ)のようにできる。この行列を用いて追加積の数を最小化すると、順列は$x_1x_2\rightarrow x_3x_4\rightarrow x_5\rightarrow x_1x_2$となる。この追加積の数は５であり、これ以上小さくはできない。この積項を積項表に加えると図７．２(ｆ)になり、対応するグラフは図７．２(ｅ)になる。関数は次のようになる。

$f'=x_1x_2\overline{x_3}+x_1x_2\overline{x_4}+x_3x_4\overline{x_5}+x_5\overline{x_1}+x_5\overline{x_2}$。

積項数は７から５に、リテラル数は１８から１２に削減した。そして、x_1x_2とx_3x_4とx_5の間の等価性を保存している。

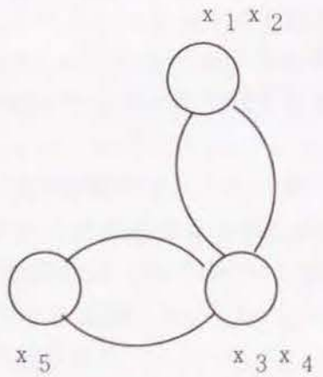
７．５　記憶最小表現

本節では、関係データベースの技術を応用した簡潔な表現法を述べる。関係データベース設計では、等価キーは一つの関係の中に記憶して、すべての関係を記憶する必要はない。ホーン節にも、同じような記法を適用する。関係データベースの等価キーに似ている等価クラスを表現するために、記憶最小表現の記法

ＩＤ（g_{k1}、g_{k2}、・・・、g_{knk}）

を使用する。このＩＤは関係名に対応し、整数は記憶コストを削減するために使用す

る。図７．２の等価クラスは、ｉ（x_5、x_1x_2、x_3x_1）と表わすことができる。この表現は、図７．２の関数ｆの中のすべての非冗長積項を含んでいて、積項数を３に、リテラル数を５に削減した。この記法には次の利点がある。



（ａ）

正リテラル部	負リテラル部
x_1x_2	$\overline{x_3}+\overline{x_4}$
x_3x_4	$\overline{x_1}+\overline{x_2}+\overline{x_5}$
x_5	$\overline{x_3}+\overline{x_4}$

積項数７、リテラル数１９

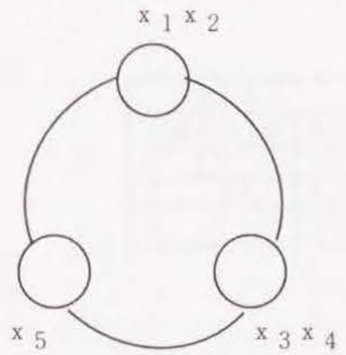
（ｂ）

	x_1x_2	x_3x_4	x_5
x_1x_2	－	2	1
x_3x_4	2	－	1
x_5	2	2	－

（ｃ）行列Ａ

	x_1x_2	x_3x_4	x_5
x_1x_2	－	$\overline{x_3}+\overline{x_4}$	$\overline{x_5}$
x_3x_4	$\overline{x_1}+\overline{x_2}$	－	$\overline{x_5}$
x_5	$\overline{x_1}+\overline{x_2}$	$\overline{x_3}+\overline{x_4}$	－

（ｄ）行列Ｂ



（ｅ）

正リテラル部	負リテラル部
x_1x_2	$\overline{x_3}+\overline{x_4}$
x_3x_4	$\overline{x_5}$
x_5	$\overline{x_1}+\overline{x_2}$

積項数５、リテラル数１３

（ｆ）

図７．２　強連結成分のループの並び順の決定

１．等価クラスを表現する記憶コストを削減する。
２．遷移的に定義される含意の情報を容易に得られる。
３．等価クラスの入力枝と出力枝のコンパクトな表現ができる。

論理関数による表現は、図７．２(a)、(e)のような選ばれた枝のみを表示する。しかし、例えば、図７．２のｆには$x_1 x_2 \rightarrow x_5$が表現されていないけれども、この記憶最小表現を用いれば容易に可能な枝すべてを生成することができる。

更に、等価クラスのＩＤは、入力枝と出力枝を表現する記憶コストを削減するのに使用できる。例えば、図７．２の含意に加えて、$x_6 \rightarrow x_1 x_2$（入力枝）と$x_3 x_4 \rightarrow x_7$（出力枝）があると仮定する。前節の手法によって最小コスト積を選択して、$x_6 \rightarrow x_5$（即ち$x_6 \overline{x_5}$）と$x_5 \rightarrow x_7$（即ち$x_5 \overline{x_7}$）で置き換える。この置換した表現は可能な含意の一部を示したのみである。等価クラスのＩＤがｉであれば、記憶最小表現は、$x_6 \rightarrow i$とｉ$\rightarrow x_7$になる。$x_6 \rightarrow i$から容易に$x_6 \rightarrow x_5$、$x_6 \rightarrow x_1 x_2$、$x_6 \rightarrow x_3 x_4$を推測できる。

７．６　評価結果

提案した手法に基づいて、知識圧縮プログラムをＣ言語を用いて作成し、Sun Sparc １０上で、ランダムに作成した２００個のルールベースに対して実験した。ルールベースの入力変数は１５以上３０未満に、積項数は２０以上３５未満とした。また、一つの積項に現われる正リテラルの数は最大４にした。

（１）巡回含意法

表２に巡回含意法を用いて積項数とリテラル数を比較した実験結果を示す。本手法が積項数を削減できたのは７０％程度あり、非冗長積和論理に対しては２０％であった。リテラル数に関して削減できたのは９０％以上に達し、非冗長積和論理に対しては２２％であった。非冗長積和論理に対して結果はよくないが、これは等価クラスが存在しないものも含まれているからである。

表７．２　巡回含意法との比較

	積項数			リテラル数		
	削減	等しい	増加	削減	等しい	増加
入力　後に　ISOP	１３４	５９	7	１８２	9	9
ISOP　後に　巡回含意法	４１	１５０	9	４５	１４３	１２
入力　後に　巡回含意法	１４１	５４	5	１８７	9	4

（２）記憶最小表現

表３に記憶最小表現との比較結果を示す。ホーン節で表現した結果との比較では、積は７２％、リテラル数９３％の例で、記憶最小表現がホーン節よりコンパクトに表現した。また、ESPRESSOにて生成してホーン節で表現した結果との比較では、積は６７％、リテラル数は８９％）の例で、記憶最小表現がホーン節の形よりコンパクトに表現

した。

表７．３　記憶最小表現との比較

	積項数			リテラル数		
	削減	等しい	増加	削減	等しい	増加
ホーン節に対する記憶最小表現	１３４	５９	7	１８２	9	9
ESPRESSOに対する記憶最小表現	４１	１５０	9	４５	１４３	１２

７．７　本章のまとめ

ゲートアレーの未配線入力等に使用するエキスパートシステムにおける知識表現の効率的記憶法として、ルールベースのルールを書き換え削減して最小の記憶量にする、次の２つの方法を提案した。

（１）巡回含意法

ルールベースを論理関数の積和形表現したホーン節の表現を削減するヒューリスティックな手法である。先ず、論理関数をＺＢＤＤを用いて非冗長論理として表現する。この表現から等価な積項を等価クラスとして選び出した後、等価クラスに基づいて積項数が最小になるように含意関係（ルール）を書き換える。特に、含意関係をグラフ表現した時の強連結成分をサイクルに書き換えることによって、表現コストを大幅に削減する「巡回含意法」を提案した。そして、実験により有効性を確かめた。

（２）記憶最小表現

論理関数の積項の等価クラスを表現するのに、関係データベースの等価キーによる表現を応用できることを示した。この表現は、積和形による表現よりも記憶コストが小さく、等価クラスから派生できる含意を列挙するのに有効である。

第８章　結論

８．１　本研究のまとめ

本論文では、標準セルとマクロセルが混在した論理ＬＳＩの自動レイアウト設計における配置配線技術について述べた。

第１章では、序論として、本論文の目的と位置付けを述べた。

第２章では、従来の論理ＬＳＩ設計の概要とその中の階層的レイアウト設計方式について述べた。自動配置処理は、普通、初期配置手法と配置改善手法の２段階で構成される。この各々で使用される代表的な手法を述べた。自動配線手法は、汎用型とモデル限定型に分類できる。この各々で使用される代表的な手法を述べた。特に、本配置配線システムに組み込んだチャネル配線法についてその基本的考え方と問題点を論じた。

第３章では、配線長最小問題の基本であるスパンニング木とスタイナ木の性質と両者の関係について述べた。更に、従来の垂直水平２方向を拡張して、配線方向の数を任意のλ（正整数）個許すλ－幾何の説明を述べた。

第４章では、λ－幾何（$\lambda \equiv 1$，２(mod 3)）のスタイナ木の性質を述べ、新たな作成法を提案した。

先ず、λ－幾何における２点間の距離の計算式を導いた。λ－幾何における２点間の距離は、ユーグリッド平面上の２点間の距離と、２点を通る直線が直近の軸方向となす角度を用いて計算することができることを述べた。

３点の最小スタイナ木に関して、$\lambda \equiv 1$、２(mod 3)の場合に、2πをできるだけ均等に分割する点としてσ点を定義して、$\lambda \equiv 1$または$\lambda \equiv 2$(mod 3)の場合、λ－幾何の３点の最小スタイナ木のスタイナ点はσ点であることを予想した。$\lambda = \infty$のと$\lambda = 2$の場合、既知の３点の最小スタイナ木のスタイナ点とσ点は一致することを示し、３点が特別な位置関係にある場合には、σ点が３点の最小スタイナ木のスタイナ点になることを証明した。

次に、この予想を用いて、２点で決まるλ－幾何の図形を用いた幾何学的手段を開発し、３点スタイナ木及び４点スタイナの作成法を開発した。λ－幾何の最小スパンニング木を出発点として、３点／４点の部分木を３点／４点のスタイナ木で置き換えることを繰り返す、与えられた点集合Ｐの多点スタイナ木の作成方法を開発した。本方法は直交幾何のLee & Sechenの方法[４３]のλ－幾何への拡張になっていることも述べた。

最後に、４－幾何のタイナ木作成法をSun Spark １０model ４０上にＣ言語で実現し、実験評価をした。評価は、乱数を用いて点座標を発生させ、点数ごとに１００回の平均値を求めた。提案手法にて作成した４－幾何のスタイナ木の枝長和と４－幾何の最小スパンニング木の枝長和を比較した結果、最小スパンニング木から３.２％短縮できた。

変更を３点木に限定した場合は３.０％短縮できた。また、従来の垂直水平配線に斜め方向の配線を追加することは配線長を１０％程短縮する大きな効果があることを示した。また、S.Burman et al.[４]の短縮率は平均８．７％であるので、本方法の方が効果が大きい。

第５章では、標準セルとマクロセルを混在できるレイアウトモデルと配置配線手法を述べた。ブロックは、階層レイアウト設計の下位階層レイアウトである。従来の自動レイアウトの対象となる標準セルはサイズ、形共に大きな差異はないが、マクロセルは標準セルと比べてサイスが数倍から数十倍大きい。先ず、この２種類のセルのサイズの差異による無駄領域を削減する拡張レイアウトモデルを提案した。拡張レイアウトモデルでは、比較的小さいマクロセルは標準セル列の中に入れ、大きいマクロセルは単独で標準セル列と同格にして高さの大きいセル列の一つとして扱う。

次に配置配線手法を述べた。まず、巨大マクロセルを核セルとして、ブロックをサブブロックに分割する手法を述べた。サブブロック内は従来の標準セルを扱う手法でセルを配置する。マクロセルは標準セルと異なり左右側面の端子にも配線する必要がある。マクロセルの側面端子の配線による無駄領域を最小に押さえる配線法を述べた。セル隙間チャネルとセル列間チャネル設けて、端子からの水平線はセル隙間チャネルのチャネル配線法で配線し、垂直線はセル列間チャネルののチャネル配線法で配線する。

最後に、実験結果により拡張モデルが標準セルとマクロセルの混在配置配線に有効であることを示した。本方法を組み込んだブロック内配置配線プログラムは、製品の論理ＬＳＩのレイアウトに使用されている。

第６章では、ブロック間配線の方法について述べた。ビルディングブロック方式でのブロックの配置は、直線によって再帰的に２分できるスライス構造配置と再帰的には２分できない非スライス構造配置に分かれる。非スライス構造配置を配線順序グラフで表現した時に現われるサイクルを単純サイクルと複合サイクルに分類して、複合サイクルはブロックを移動して単純サイクルにして、単純サイクルの非スライス構造配置を配線する「単純サイクル巡回配線法」を提案した。本方法は、チャネルグラフ上の４つの頂点からなるサイクルに対応するチャネルを巡回して、チャネルの接合部の座標値が一致するまで繰り返し配線する。何回も繰り返し配線しても一致しない時に繰り返し回数を抑さえるための、３辺固定チャネル配線も提案した。実験結果は、繰り返し配線のみで十分であり３辺固定チャネル配線は不要であった。これにより本方法が有効であることを確認した。

また、チャネル配線法を拡張して、信号配線と同時に電源線を幅広配線する方法を開発した。本方法を組み込んだブロック間配線プログラムは、製品の論理ＬＳＩのレイアウトに使用されている。

第７章では、ゲートアレーの未配線等を扱うルールベースのエキスパートシステムの大量の知識を効率良く記憶する手法について述べた。エキスパートシステムはプロダク

ションルールの各ルールは論理関数で表わすことができる。ルールベースを論理関数の積和形で表現し、ルールベースのルールを書き換え削減して最小の記憶量にする、次の２つの方法を提案した。

（１）巡回含意法：ルールベースを論理関数の積和形表現したホーン節の表現を削減するヒューリスティックな手法である。先ず、論理関数をＺＢＤＤを用いて非冗長論理として表現する。この表現から等価な積項を等価クラスとして選び出した後、等価クラスに基づいて積項数が最小になるように含意関係（ルール）を書き換える。特に、含意関係をグラフ表現した時の強連結成分をサイクルに書き換えることによって、表現コストを大幅に削減する。そして、実験により有効性を確かめた。

（２）記憶最小表現：論理関数の積項の等価クラスを表現するのに、関係データベースの等価キーによる表現を応用できることを示した。この表現は、積和形による表現よりも記憶コストが小さく、等価クラスから派生できる含意を列挙するのに有効である。

８．２　今後の課題

１９７０年代より研究が開始されてきたレイアウト問題は、現在では成熟分野になってきた。しかし、論理ＬＳＩは、ますます高速化、大規模化が進展している。レイアウト結果のより一層の高品質化、高密度化という困難な課題が依然として残されている。

高品質化のためには、信号伝播における配線遅延時間を最適化するレイアウトを実現することである。同期回路では、第１に、信号のパスディレイの問題がある。パスディレイを求めるためには、先ず、ゲートごとのディレイを計算しなければならない。ゲートを集中定数回路モデルで表わし、配線長の負荷を加えて、ゲートのディレイを計算する。これをパスに従って積算してパスディレイとし、セルの配置および配線経路を決定する必要がある。どうしても配線長が長くなる場合には、中間に２段のバッファを挿入して論理までも変更する必要がある。第２に、クロック信号やバス信号のスキューの問題がある。クロック信号のスキューを最小にするために、クロック発生器からフィリップフロップやクロックゲートまでのディレイが等しくなるように、フィリップフロップやクロックゲートを配置し配線する必要がある。また、バス信号も、バス信号内の各信号のスキューを最小にする必要がある。本論文の第６章において、バス信号は同じブロック間を経由する配線経路にする方法を述べた。しかし、人手配線する場合のように定常流の流線のように並んだ配線にする必要がある。これらの問題に対応するためには、配置配線手法の機能追加が必要なことはもちろんであるが、レイアウト設計のみでは解決できず、論理設計、回路設計と連係して、レイアウト設計結果をチェック後に論理設計、回路設計フェーズに戻るという大局的な設計が必要になって来ている。

また、大きな電流が流れる電源線やクロック線は電流変化がクロストークノイズの原因になるので、他の信号線と離した位置に配線する必要がある。そして、配線の抵抗値を小さくするためとマイグレーションを防止するために、配線幅が広い配線をする必要がある。幅広配線については、本論文の第６章において述べた電源線の自動配線方法を応用できる。しかし、更にきめ細かい扱いが必要になって来ている。

高密度化のためには、規則的な構造を持つＡＬＵ、乗算器や、単純な繰り返し構造のＲＡＭ、ＲＯＭなどは、人手で作成した方が密度が高いレイアウトができ、自動設計が遅れているという問題がある。このためには、トランジスタレベルの配置配線技術を開発する必要がある。人手設計した回路をマクロセルとして配置配線する方法は本論文の第５章で述べた。しかしマクロセルは高密度に小さくレイアウトすると常に矩形になるとは限らない。矩形以外の直角多角形になることがある。このため、直角多角形を扱うことができて、且つ、無駄な領域を発生させない配置配線技術が必要である。

一方、日々進歩している製造プロセス技術に合わせた配置配線技術を開発する問題もある。具体的には、多層配線が可能になり、現在では、５～６層配線も現実のものとなりつつある。配線層数が増加すると、論理ＬＳＩの高速化のために、プリント基板と同様に斜め配線が採用される可能性がある。斜め配線が採用されれば、配線長最小問題としてのスタイナ木は、本論文第４章で述べたλ－幾何スタイナ木の$\lambda=4$の場合に対応する。この問題に対して本論文のスタイナ木作成法を応用することができる。

謝　辞

本研究の推進、および、本論文の執筆に関して以下の各氏に深甚の謝意を表わします。最初に、知識の記憶最小化の研究の御指導を頂き、更には、本論文をまとめるに当たり貴重な時間を割いてきめ細かい御指導を頂いた京都大学工学部情報工学科上林弥彦教授に深く御礼申し上げる。

また、岡山県立大学情報工学部学部長本田和男教授、情報システム工学科長弓場芳治教授には、研究に専念できる機会を与えられ、暖かく見守って頂いたことに対して心からの感謝を申し上げる。

小澤時典副技師長、寺井秀一部長（現在、立命館大学工学部電気電子学科教授）には、（株）日立製作所中央研究所在籍の時、直接の上司として、論理ＬＳＩの自動設計の研究を始める端緒を作って頂き、その後も論理ＬＳＩの自動設計の研究を遂行するに当たって終始暖かい御指導を頂いた。深く感謝の意を表わす。

また、論理ＬＳＩの自動配置配線システムを開発するに当たって共に努力し苦労した石井建基主任技師、三浦地平主任研究員、そして、論理ＬＳＩの自動設計システムの仕様に関して御指導を頂いた半導体事業部ＤＡ開発部山城治部長、秋山俊恭主任技師に心からの感謝を申し上げる。

上林弥彦教授の下で知識の記憶最小化の研究を共同研究をした京都大学学生永井裕氏に心からの感謝を申し上げる。

参考文献

[1] T.Asano, T.Kitahashi, K.TanakaH.Horino and N.Amano, "A Wire-Routing Scheme Based on Trunk-Division Methods,"
IEEE Trans. on Comp., vol.C-26, no.8, pp.764-772, (1977).

[2] N.K.Bose and J.H.Lee, "Steiner's Problem and its Use in Suboptimal Routing",
Proc. of 9th Annu. Asilomar Conf. on Circuits, Systems and Computers, pp.48-53, (1975).

[3] M.A.Breuer, "A Class of Min-Cut Placement Algorithms,"
Proc. of 14th Dsign Automation Conf., pp.284-290, (1977).

[4] S.Burman, H.Chen and N.Sherwani, "Improved Global Routing using λ -Geometry",
Proc. of 29th Annual Allerton Conf. on Communications, Computing and Controls, pp.1083-1092, (1991).

[5] M.Burstein and R.Pelavin, "Hierarchical Wire Routing ,"
IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., vol.CAD-2, no.4, pp.223-234, (1983).

[6] M.Burstein and M.N.Youssef, "Timing Influenced Layout Design,"
Proc. of 22nd Dsign Automation Conf., pp.124-130, (1985).

[7] M.J.Ciesielski, "Two Dimensional Routing for the Silc Silicon Compiler,"
IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., vol.CAD-4, no.3, pp.198-203, (1985).

[8] H.H.Chen, "Routing L-Shaped Channels in Non-slicing Structure Placement,"
Proc. of 24th Dsign Automation Conf., pp.152-158, (1987).

[9] T.Chiba, N.Okuda and T.Kambe, "SHARPS: A Hierarchical Layout System for LSI,"
Proc. of 18th Dsign Automation Conf., pp.820-827, (1981).

[10] B.W.Colbry and J.Soukup, "Layout Aspect of VLSI Microprocessor Design,"
Proc. of ISCAS'82, pp.1214-1228, (1982).

[11] W.Dai, T.Asano and E.S.Kuh, "Routing Region Definition and Ordering Scheme for Buiding-Block Layout,"IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., vol.CAD-4, no.3, pp.189-197, (1985).

[12] D.Deutsch, "A Dogleg Channel Router,"
Proc. of 13rd Dsign Automation Conf., pp.425-433, (1976).

[13] D.Z.Du and F.K. Hwang, "A Proof of Gilbert-Pollak Conjecture on Steiner Ration",
Proc. of 31st Annual Symp. of Foundation of Computer Science, pp.76-85, (1990).

[14] C.K.Erdelyi et al., "A Comparison of MIXED Gate Array and Custom IC Design Method," Proc.of ISSCC Digest of Technical Papers, pp.14-15, (1984).

[15] M.Fukui , A.Yamamoto, R.Yamaguchi, S.Hayama and Y.Mano, "A Block Interconnection Algorithm for Hieralchical Layout System,"IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., vol.CAD-6, no.3, pp.383-391,

(1987).
[16] T.Fujita and S.Goto, "A Rule-Based Routing System",
Proc. of Int. Conf. on Comput. Design '83, pp.451-454, (Jun. 1983).
[17] M.R. Garey, R.L.Graham and D.S. Johnson,"The Complexity of Computig Steiner Minimal Tree", SIAM J. Appl. Math., vol.32, no.4, (1977).
[18] M.R. Garey and D.S. Johnson ,"The Rectilinear Steiner Tree Problem is NP-complete", SIAM J. Appl. Math., vol.32, no.4, pp.37-58, (1977).
[19] P.L.Hammer and A.Kogan ,"Horn Functions and their DNFs",
Information Processing Letters, vol.44, no.1, pp.23-29, (1992).
[20] P.L.Hammer and A.Kogan ,"Knowledge Compression - Logic Minimization for Expert Systems", Proc. of IISF/ACM Japan Int. Symp., pp.306-312, (1993).
[21] M.Hannan ,"On Steiner's Problem with Rectilinear Distance",
SIAM J. Appl. Math., vol.14,no.2, pp.255-265, (1977).
[22] A.Hashimoto J.Stevens, "Wire Routing by Optimizing Channel Assignment within Large Aperture," Proc. of 8th Dsign Automation Workshop, pp.155-169, (1971).
[23] D.W.Hightower, "A Solution to Line-Routing Problems on the Continuous Plane," Proc. of 6th Dsign Automation Workshop, pp.1-24, (1969).
[24] D.W.Hightower, "A Generalized Channel Router",
Proc. of 17th Dsign Automation Conf., pp.12-21, (1980).
[25] J.M. Ho , G.Vijayan and C.K. Wong , "A New Algorithm for the Rectilinear Steiner Tree Problem", IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., vol.9, no.2, pp.185-193,(1990).
[26] C.P.Hsu, "A New Two-Dimensional Routing Algorithm,"
Proc. of 19th Dsign Automation Conf., pp.46-50, (1982).
[27] C.P.Hsu, L.Grate, C.Ng, M.Hartoog and D.Bohm "The Chip Compiler, An Automated Cell/Macrocell Physical Design Tool," Proc. of CICC, pp.488-491, (1987).
[28] F.K.Hwang ,"On Steiner Minimal Trees with Rectilinear Distance",
SIAM J. Appl. Math., vol.30, no.1, pp.104-114,(1976).
[29] F.K.Hwang ,"An O(n log n) Algorithm for Suboptimal Rectilinear Steiner Trees", IEEE Trans. on Circuits & Syst., vol.26, no.1, pp.177-182,(1979).
[30] F.K.Hwang ,"An O(n log n) Algorithm for Rectilinear Minimum Spanning Trees", J. Assoc. Comput. Mach., vol.26, no.2, pp.75-77,(1979).
[31] Y.Kambayashi ,"Equivalent Key Problem of the Relational Databese Model", Proc. of Int. Conf. on Math. Studies if Infor.Processing, Lecture Notes in Computer Science 75, Springer-Verlag, pp.155-181,(1978).
[32] R.Kasai et al., "An Integrated Modular and Standard Cell IC Design Method," Proc.of ISSCC Digest of Technical Papers, pp.12-13, (1984).
[33] H.Kawanishi S.Goto and T.Oyamada, "A Routing Method of Building Block LSI," Proc.of 7th Asilomar Conf. on Circuits, Systems and Computers, pp.119-123, (1973).

[34] B.W.Kernighan, D.G.Schweikert and G.Persky, "An Optimum Channel-Routing Algorithm for Polycell Layout of Integrated Circuits,"
Proc. of 10th Dsign Automation Workshop, pp.50-59, (1973).
[35] S.Kimura, T.Chiba and I.Nishioka, "An Automatic Routing Scheme for General Cell LSI," IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., vol.CAD-2, no.4, pp.285-292, (1983).
[36] T.Kozawa, C.Miura and H.Terai, "Combine and Top Down Block Placement Algorithm for Hierarchical Logic VLSI Layout," Proc. of 21st Dsign Automation Conf., pp.667-669, (1984).
[37] J.B.Kruscal, "On the Shortest Spanning Subtree of a Graph,"
Proc. of Ammer. Math. Soc., vol.7, pp.48-50, (1956).
[38] N.Kuwahara, S.Asami, N.Tanaka and M.Nomura , "A Routing System for High-Performance Computer Systems", Proc. Int. Conf. on Comput.-Aided Design '86, pp.250-253, (1986).
[39] U.Lauther, "Channel Routing in a General Enviroment,"
Proc. of VLSI Symp.85, pp.389-399, (1985).
[40] U.Lauther, "A Min-Cut Placement Algorithm for General Cell Assemblies Based on a Graph Representation," Proc. of 16th Dsign Automation Conf., pp.1-10, (1979).
[41] C.Y.Lee, "An Algorithm for Path Connections and its Application,"
IRE Trans. on Erectronic Computers, vol.EC-10, pp.346-365, (1961).
[42] J.H.Lee and F.K. Hwang, "Use of Steiner's Problem in Suboptimal Routing in Rectilinear Metric", IEEE Trans. on Circuits & Syst., vol.CAS-23, no.7, pp.470-476,(1976).
[43] K.Lee and C.Sechen , "A New Global Router for Row-Based Layout",
Proc. Int. Conf. on Comput.-Aided Design '88, pp.180-183, (1988).
[44] W.Liu and D.E.Atkins, "On the Routability and Routing Order of a General Cell Approach," Proc. of Int. Conf. on Circuits Comput. '82, pp.246-249, (1982).
[45] W.K.Luk, D.T.Tang and C.K.Wong, "Hierarchical Global Wiring for Custom Chip Design," Proc. of 23rd Dsign Automation Conf., pp.481-489, (1986).
[46] K.Mikami and K.Tabuchi, "A Computer Program for Optimal Routing of Printed Circuit Connectors," Proc.of IFIPS, vol.H47, pp.1475-1478, (1968).
[47] B.T.Preas and C.W.Gwyn, "Methods for Hierarchical Automatic Layout of Custum LSI integrated Circuits," Proc. of 15th Dsign Automation Conf., pp.206-212, (1978).
[48] R.C.Prim,"Shortest Connection Networks and Some Generalizations ",
Bell Sys.Tech.j., vol.36, pp.1389-1401, (1957).
[49] J.Reed, A.Sangiovanni-Vincentelli and M.Santomauro, "A New Symbolic Channel Router: YACR2," IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., vol.CAD-4, no.3, pp.208-219, (1985).
[50] R.L.Rivest and C.M.Fiduccia, "A "Greedy" Channel Routert,"

Proc. of 19th Dsign Automation Conf., pp.418-424, (1982).

[51] M.Sarrafzadeh & C.K.Wong ,"Hierarchical Steiner Construction in Uniform Orientations", IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., vol.11, no.9, pp.1095-1103,(1992).

[52] M.Sarrafzadeh & C.K.Wong ,"Bottleneck Steiner Trees in the Plane", IEEE Trans. on Computers, vol.41, no.3, pp.370-374,(1992).

[53] K.Sato, T.Nagai, M.Tachibana, H.Shimoyama, M.Ozaki and t.Yahara, "MILD-A Cell Based Layout System for MOS-LSI," Proc. of 18th Dsign Automation Conf., pp.828-836, (1981).

[54] Y.Sekiyama, Y.Fujiwara,T.Hayashi, M.Seki, J.Kusuhara, K.Iijima, M.Takahara and K.Fukatani, "Timing-Oriented Routers for PCB Layout Design of High-Performance Computers", Proc. of Int. Conf. on Comput.-Aided Design '91,pp.332-335,(1991).

[55] N.Sherwani, *Algorithms for VLSI Pysical Design Automation, 2nd. ed.*, Kluwer Academic Publishers, (1995).

[56] D.M.Schuler and E.G.Ulrich, "Clustering and Linear Placement," Proc. of 9th Dsign Automation Workshop, pp.50-56, (1976).

[57] T.Yoshimura and E.S.Kuh, "Efficient Algorithms for Channel Routing," IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., vol.CAD-1, no.1, pp.25-35, (1982).

[58] P.Windmayer ,Y.F. Wu & C.K. Wong , "On Same Distance Problem in Fixed Orientations", SIAM J. Comput., vol.16, no.4, pp.728-746(1987).

[59] P.Winter ,"Steiner Problem in Netwoks: A Survey", Networks, vol.17, pp.129-167,(1987).

[60] 秋山仁、R.L.Graham 「離散数学入門」朝倉書店 (1993).

情報処理設計自動化研究会、44-8, pp.59-65, (1988).

[62] 堀野、北爪、平野、" ＬＳＩ素子間列間の配線手法"
電子通信学会研究会資料、SSD-70-64, (1971).

[63] 三浦　他、" ＶＬＳＩブロック自動配置手法"
情報処理設計自動化研究会、24-2, pp.1-8, (1984).

[64] 光安　他、" スタンダードセル自動レイアウトシステム：STELLA 自動配線システム" 電子通信学会研究会資料、vol.86, no.148, pp.9-16, (1986)

[65] 湊、" 二分決定グラフからの非冗長積和論理の高速生成手法"
情報処理設計自動化研究会、91-DA-60, pp.147-154, (1991).

発表文献

1．主要論文

[66] 早瀬、平野、武市　" ページ方式による多数ユーザ会話システムの解析"
電子通信学会誌 vol.57-D, no.3, pp.105-111 (Mar. 1974).

[67] H.Terai, M.Hayase, T.Ishii, C.Miura, T.Kozawa, K.Kishida, and Y.Nagao, "Automatic Placement and Routing Program for Logic VLSI Design based on Hierarchical Layout Method,"
Proc. of Int. Conf. on Circuits & Comput.'82, pp.415-418, (Sep. 1982).

[68] T.Kozawa, H.Terai, T.Ishii, M.Hayase, C.Miura, Y.Ogawa, K.Kishida, N.Yamada and Y.Ohno, "Automatic Placement Algorithms for High Packing Density VLSI," Proc. of 20th Design Automation Conf., pp.175-181, (Jun. 1983).

[69] H.Terai, M.Hayase, T.Ishii, C.Miura, Y.Ogawa, Y.Kozawa, Y.Sato and Sasaki, "Performance Anaiysis of Automatic Placement and Routing for Large-Scale CMOS Masterslicees",
Proc. of Int. Conf. on Comput. Design '83, pp.536-539, (Jun. 1983).

[70] H.Terai, M.Hayase and T.Kozawa, "A Routing Procedure for Mixed Array of Custom Macro and Standard cells," Proc. of 22nd Design Automation Conf., pp.503-508, (Jun. 1985).

[71] H.Nagai, M.Hayase and Y.Kambayashi, "Storage Minimum Expressions for Horn Clauses Utilizing Binary Decision Diagrams," Proc. of 5th European-Japanese Seminor on Information Modelling and Knowledge Bases, (Jun. 1995).

[72] H.Nagai, M.Hayase and Y.Kambayashi, "Minimization and Storage Minimum Expressions for Horn Clauses of Propositional Logic,"
IOS press (to appear 1995).

[73] M.Hayase, "A New Algorithm to Construct Steiner Minimal Trees in Uniform Orientations,"
Iiformation Letters (submitted).

2．研究会及び大会資料

[74] 早瀬、上林　" Tree Transducer について"
信学会　オートマトン研究会資料 A69-42, (1969).

[75] 小澤、石井、三浦、小川、寺井、早瀬、岸田、山田　" ＶＬＳＩ配置手法の評価"
情処　設計自動化研究会報告 18-4, (1983).

[76] 寺井、早瀬、石井、三浦、小澤、佐藤、桧山、" 大規模ＣＭＯＳマスタスライス自動配線方式"
信学会　回路とシステム研究会資料 CAS83-2111, (1983).
[77] 早瀬、小澤、寺井、秋山、福田、" マクロセルを含むＶＬＳＩブロックの自動配線方式"
情処　設計自動化研究会報告 vol.87,no.74, 87-DA-39-8, (Oct. 1987).
[78] 早瀬、三浦、山城、" ブロック間配線における非スライス構造配置の一配線法"
情処　設計自動化研究会報告 vol.89,no.14, 89-DT-46-3, (Feb. 1989).
[79] 上林、早瀬、" 移動分散データベースにおける質問処理"
情処　データベース研究会報告 vol.104,no.20, pp.153-160, (Jul. 1995).
[80] 上林、早瀬　" Tree Transducer について"
信学全大 39、(Mar. 1969).
[81] 早瀬、武市、津田、平野、" オンラインシステムの応答時間の一計算法"
信学全大 1373, (Mar. 1973).
[82] 寺井、石井、早瀬、小澤、湯山、長尾　" ＬＳＩセルの一配置法"
信学全大 550, (Mar. 1981).
[83] 早瀬、石井、寺井、小澤　" 信号遅延を考慮した３層配線プログラムの開発"
信学全大 332, (Mar. 1982).
[84] 石井、早瀬、寺井、小澤、三浦、長尾、湯山　" ＶＬＳＩセル自動配置プログラムの処理方式"
信学全大 346, (Mar. 1982).
[85] 三浦、小川、石井、寺井、早瀬、小澤、佐々木　" 可変長セルの２次元初期配置手法"
信学全大 347, (Mar. 1982).
[86] 寺井、早瀬、石井、三浦、小澤、杉山、山田　" 階層設計向きＶＬＳＩセル自動配置配線
プログラムの開発"　信学全大 348, (Mar. 1982).
[87] 石井、早瀬、寺井、小澤、小川、佐々木、佐藤、桧山　" ＶＬＳＩ自動配置配線プログラム
の開発（ALEPH配置配線システムの構成）"　情処全大 5K-5, (Mar. 1983).
[88] 早瀬、寺井、石井、小澤　" 両面端子セルの端子割付手法（ALEPH配線手法－１）"
情処全大 5K-8, (Mar. 1983).
[89] 白石、早瀬、石井、小澤　" チャネル内配線における幹線分割法（ALEPH配線手法－2）"
情処全大 5K-9 ,(Mar. 1983).
[90] 佐藤、岸田、高亀、池本、早瀬、千葉　" 複数アルゴリズムを組み合わせた配線手法（ALEPH配線手法－3）"
情処全大 5K-10, (Mar. 1983).

[91] 三浦、石井、小川、小澤、寺井、早瀬、佐々木、桧山　" 正規化された面積値を用いた配置
アルゴリズムの比較評価"　信学全大 437, (Mar. 1983).
[92] 早瀬、秋山、福田　" マクロセルと標準セル混在自動配置手法"
信学全大 410, (Mar. 1985).